

AFIT/GE/ENG/03-07



THEORETICAL ANALYSIS OF INFORMATION WATERMARKING IN  
WAVELET-BASED VIDEO COMPRESSION

THESIS  
Brendan Kevin Casey  
Captain, USAF

AFIT/GE/ENG/03-07

DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY

***AIR FORCE INSTITUTE OF TECHNOLOGY***

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense or U.S. Government.

AFIT/GE/ENG/03-07

THEORETICAL ANALYSIS OF INFORMATION  
WATERMARKING IN WAVELET-BASED VIDEO  
COMPRESSION

THESIS

Presented to the Faculty  
Graduate School of Engineering and Management  
Air Force Institute of Technology  
Air University  
Air Education and Training Command  
In Partial Fulfillment of the Requirements for the  
Degree of Master of Science in Electrical Engineering

Brendan Kevin Casey, M.S. E.S.M., B.S. E.E., B.S. C.E.  
Captain, USAF

March 2003

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

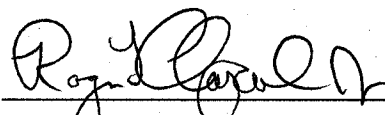
THEORETICAL ANALYSIS OF INFORMATION  
WATERMARKING IN WAVELET-BASED VIDEO  
COMPRESSION

THESIS

Brendan Kevin Casey, M.S. E.S.M., B.S. E.E., B.S. Co.E.

Captain, USAF

Approved:

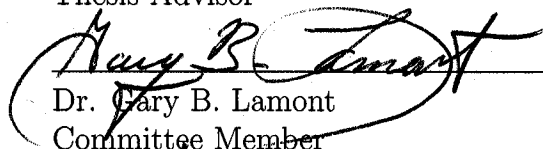


Maj Roger L. Claypoole, Jr., Ph.D.

Thesis Advisor

6 MAR 03

Date

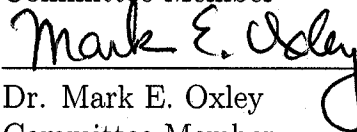


Dr. Gary B. Lamont

Committee Member

6 MAR 03

Date



Dr. Mark E. Oxley

Committee Member

6 March 03

Date

## *Acknowledgements*

I'd like to thank Maj Claypoole as my thesis advisor. He allowed me to follow my own chaotic schedule and brought me back on track when necessary. I'd also like to thank Dr. Oxley and Dr. Lamont who granted me wide latitude to work within. Capt Hale, a true LaTeX guru, also deserves thanks.

Thanks to friends like Gary and Charlotte who stalked me from Texas and summer camp. Stalked in a nice way, of course. And thanks to Bryan who followed me from the squadron to the school. To Courtney, Erin, Randy, and Nicole, thanks for the crazy cosmic bowling nights amid other outings. Thanks to my fellows in the VIP lab, Kate, Dan, and Andy who refused to allow anyone to get more than 30 minutes of work done at a time. Joe, who introduced me to making pizzas from scratch is owed a big debt of thanks, as are the bi-weekly Friday night Halo crew who aided in stress relief. Finally, to everyone else who deserves a thank you, I thank you too.

Brendan Kevin Casey

# Table of Contents

	Page
Acknowledgements . . . . .	iv
List of Figures . . . . .	vii
List of Tables . . . . .	viii
Abstract . . . . .	ix
I. Introduction . . . . .	1-1
1.1 Problem Statement . . . . .	1-1
1.2 Scope . . . . .	1-1
1.3 Organization . . . . .	1-2
II. Background . . . . .	2-1
2.1 Human Visual System . . . . .	2-1
2.2 Measuring Image Quality . . . . .	2-2
2.3 Wavelets . . . . .	2-3
2.3.1 Multi-Decomposition . . . . .	2-3
2.3.2 Parsimony . . . . .	2-6
2.4 Embedding Audio/Video . . . . .	2-7
2.5 Watermarking . . . . .	2-9
2.6 Quantization/Reconstruction . . . . .	2-10
2.6.1 Bit Plane Coding . . . . .	2-10
2.6.2 Index Coding . . . . .	2-11
2.7 Mendenhall's Digital Watermarking System . . . . .	2-13
2.8 Bose-Chadhuri-Hocquenghem Error Correcting Code . . . . .	2-15
2.9 Summary . . . . .	2-16
III. Methodology and Design . . . . .	3-1
3.1 Two Channel Structure . . . . .	3-1
3.2 Variables . . . . .	3-2
3.3 Measurements . . . . .	3-5
3.3.1 Peak Signal to Noise Ratio . . . . .	3-6
3.3.2 Information Bit Error Rate (IBER) . . . . .	3-6
3.3.3 Bit Rate . . . . .	3-6
3.4 Quantization . . . . .	3-7
3.5 Regions on the IBER plot . . . . .	3-9

	Page
3.5.1 Dropped Bits region . . . . .	3-9
3.5.2 Extraction Error region . . . . .	3-10
3.5.3 Perfect Extraction region . . . . .	3-10
3.5.4 Theoretical Result . . . . .	3-10
3.6 Summary . . . . .	3-12
IV. Results and Analysis . . . . .	4-1
4.1 Given IBER acceptable to the user, T, and Q, choose S . . . . .	4-1
4.2 Given T, Q, and S, choose $N_Q$ . . . . .	4-5
4.3 Given T, S, and N, choose an Error Correction Code to meet the required IBER . . . . .	4-8
4.3.1 Choosing N, n, k, and M . . . . .	4-9
4.3.2 Binary Symmetric Channel . . . . .	4-10
4.3.3 Error Correction Code Results . . . . .	4-12
4.4 Given T, choose an N, Q, and S to give the desired PSNR . . . . .	4-14
4.5 Example Output . . . . .	4-17
4.6 Summary . . . . .	4-19
V. Conclusions . . . . .	5-1
5.1 Contributions . . . . .	5-1
5.2 Future Work . . . . .	5-3
Bibliography . . . . .	BIB-1
Vita . . . . .	VITA-1

# List of Figures

Figure		Page
2.1.	Original 256 x 256 8-bit gray scale “Lenna” image. . . . .	2-4
2.2.	The original image decomposed into four subbands. . . . .	2-5
2.3.	A simplified example of the two-dimensional wavelet transform. . . . .	2-5
2.4.	The original “Lenna” image after three iterations of the wavelet transform. . . . .	2-6
2.5.	Plot of the logarithmic of the number of wavelet coefficients with the given energy magnitude. . . . .	2-7
2.6.	Example of parsimony property of wavelet transform used in image compression. . . . .	2-8
2.7.	Scanning order to process coefficients. . . . .	2-12
2.8.	The five sections of Mendenhall’s Digital Watermarking System embedding an audio bit stream into the image frame. . . . .	2-14
3.1.	Two Channel representation of watermarking and compression system. . . . .	3-1
3.2.	Bin layout based on $T = 11$ and $Q = 8$ . . . . .	3-8
3.3.	Bins with reconstruction values for $T = 11$ and $Q = 8$ . . . . .	3-8
3.4.	Three Regions of Information Bit Error Rate versus $S$ using experimental results. . . . .	3-9
3.5.	Three Regions of Information Bit Error Rate versus $S$ using theoretical calculations. . . . .	3-11
4.1.	Bin edges, reconstruction values, and embedded bit values. . . . .	4-2
4.2.	Bin edges, reconstruction values, and embedded bits values for $S = \frac{3}{4}t_o$ . . . . .	4-3
4.3.	Bin edges, reconstruction values, and embedded bits values for $S = \frac{t_o}{2}$ . . . . .	4-4
4.4.	Example of how the modulo arithmetic changes the acceptable minimum coefficient over a range of $S$ . . . . .	4-7
4.5.	For $T = 11$ and $Q = 8$ , the system is binary symmetric for most values of $S$ . . . . .	4-11
4.6.	Plots showing the results of using BCH error correction codes with $T = 11$ and $Q = 8$ but different $n, k$ combination. . . . .	4-13
4.7.	The impact on PSNR of varying the number of image coefficients available for reconstruction. . . . .	4-15
4.8.	Contour lines of constant PSNR for the varying $S$ and $Q$ . . . . .	4-16
4.9.	For a large constant $N$ , PSNR varies over the range of $S$ and $Q$ . . . . .	4-17
4.10.	Sample example of system. . . . .	4-18



# *List of Tables*

Table		Page
4.1.	<i>Algorithm for determining <math>N_Q</math> . . . . .</i>	4-7
4.2.	<i>Overhead cost for using BCH error correcting code. . . . .</i>	4-12

## *Abstract*

Embedding audio bits into images for transmission of video data alleviates the synchronization problem common in video transmission techniques. We continue work combining audio or other information bits and images into one file using digital watermarking techniques to correct the synchronization problem. The system compresses the file by using wavelet image coefficients and implementing bit plane coding.

Our research encompasses incorporating five free variables into the watermark/compression technique. These variables are watermark robustness, number of coding iterations, number of image coefficients, number of watermarked information bits, and number of watermarked error correcting bits. By altering these variables, four measurements of the output change. The measurements are the information bit error rate, the image quality, the bit rate, and the amount of watermarked data. We mathematically demonstrate how the variables impact these measurements. Experimental results on real video data support our findings. By analyzing each video frame, an automated system is able to choose optimal values of the five variables to meet specified measurement constraints.

# THEORETICAL ANALYSIS OF INFORMATION WATERMARKING IN WAVELET-BASED VIDEO COMPRESSION

## *I. Introduction*

### *1.1 Problem Statement*

With today's heightened security concerns, smaller deployed forces are desired. Reach back capabilities allow deployed units access to information required while putting fewer people into danger. With fewer people deployed, a smaller area is required to station the deployed force. Securing a smaller physical area has manpower, economic, and political savings. The deployed force relies upon the reach back units stationed in the continental United States to perform vital functions the deployed force cannot execute. Communication between the force and units is necessary for mission success. Communication between the two must be information rich which leads to video data.

Video data is a combination of images and audio and requires a large bandwidth to transmit. It is possible to directly combine the audio/video and compress both simultaneously.

### *1.2 Scope*

This research further demonstrates that the technique pioneered by Mendenhall [10] is a viable solution to transmitting video data. In his work, he embedded the audio information into the image frame using digital watermarking techniques and wavelet transform as opposed to appending the information at the end of the frame as Zhang and Zheng [22] propose at Ohio State University. He then compressed the

resulting wavelet coefficients. We further this work by incorporating error correction code that increases the reliability of the watermarked information without detracting from the image quality.

We increase the versatility of this technique by introducing variables that control the state of the system. These variables control the number of wavelet coefficients to use, the strength of the embedded watermark, and the number of iterations to perform during quantization, among others. By adjusting these variables in a known manner, we are able to achieve user specified requirements for the state of the system. These requirements entail specifying an error rate in the information bits transmitted, an amount of information to send, a bit rate to transmit, and an image quality. This allows the system to be flexible under different situations.

We also expand this research to embed any binary information. No restriction exists on the type of binary data to embed as the information.

### *1.3 Organization*

This thesis is divided into four chapters. Chapter II provides background information on different topics relevant to this work. This background will aid in better understanding of Chapter III where we explain what the variables are, what the measurements are, and the specifics of the system. We present the results and analysis of this thesis research in Chapter IV and conclude in Chapter V.

## II. Background

This chapter provides background in order to better understand the rest of this thesis. The background begins with a description of the Human Visual System followed by a description of image quality measurement. Some properties of wavelets are explained; wavelets are used in this study to embed the audio information into the image files as a digital watermark and for transform-based compression. This technique is based on a previous algorithm derived by Mendenhall; that work is also explained. We conclude with a discussion of error correction codes.

### 2.1 Human Visual System

Any image processing system must take into account the Human Visual System (HVS). The HVS is composed of the eye, the optical nerve, and the brain [21]. Within the eye, the retina is composed of rods and cones. The rods perform better in low light, and so deal mostly with colorless, gray scale images. The cones are divided into three types each sensitive to a different frequency, color, of light. The information from the rods and cones is sent along the optical nerve to the brain to be processed. Each part of the HVS can introduce error to the final image processing. Because the HVS is not a perfect system, the image processing system must simply reconstruct an image that meets the detection criteria of the HVS. The reconstructed image does not need to be an identical copy but one that is recognizable. That is, the reconstructed image can suffer from errors that are not noticeable to the human eye.

Because the reconstructed image can contain these non-noticeable errors, noise or data can be inserted into the reconstructed image without loss of recognition by the HVS [19]. Insertion of this data is called visual masking. Three forms of visual masking exist: *spatial*, *spectral*, and *temporal* masking [6,14].

Spatial masking uses the luminance contrast of an image to conceal data. Noise added to a highly textured image blends with the already highly contrasting luminance of the image. In a smooth image, less contrast exists. Because of this, the

luminance contrast of the noise is pronounced against the low contrasting image. Therefore, with a highly textured image, noise or data can be added directly to the image without any noticeable change to image quality.

Spectral masking relies upon the spectral frequencies, or colors, of the image. The spectral frequencies analyzed by the HVS can be decomposed into three. They are red, green, and blue. At the lowest frequency, blue, the sensitivity of the HVS is much lower than the other two [5, 11]. For a colored image, we want to mask our information in the blue region.

Temporal masking depends upon the frequency of displayed information. This is called the flicker rate. Experiments show the sensitivity of the HVS to rates above 30Hz is very small. Sensitivity above 60Hz is about zero [7]. Standard video uses a flicker rate of 30 frames a second while movies and computer monitors use 60Hz.

In this research, we are working with black and white video images. Thus, spectral masking has no impact on our masking decisions. Because we analyze only one frame at a time, temporal masking does not concern us. We are only concerned with spatial masking. Clearly, such masking degrades image quality. We now discuss how to measure the quality of an image.

## 2.2 *Measuring Image Quality*

Peak Signal to Noise Ratio (PSNR) is the industry standard to measure the quality of images. Equation 2.1 shows the calculation of PSNR, in decibels, with  $x$  being the original pixel value,  $x'$  being the reconstructed pixel value, and having  $N$  total pixels.

$$PSNR = 20 \log_{10} \left( \frac{\max \text{ pixel value}}{\sqrt{\sum_i^N (x_i - x'_i)^2 / N}} \right) \quad (2.1)$$

The mean squared difference for all the pixels is calculated in the denominator. The advantage of the PSNR calculation versus using just the signal to noise ratio (SNR) is with the PSNR, we take into account the maximum valued pixel in the numerator.

This normalizes the PSNR for a class of images. Because the numerator is the max value for the class, all 8-bit gray scale images are normalized the same whether they are ‘bright’ or ‘dark.’ This removes from consideration the power of the image. Power is not a consideration by the HVS in measuring quality. A more powerful image does not necessarily mean it is a higher quality image, simply brighter. By normalizing the measurement, the PSNR removes this power consideration.

### 2.3 Wavelets

Wavelet transforms are an excellent tool for image compression, providing high quality (PSNR) and an efficient representation [1]. The wavelet transform takes the image from the spatial domain and converts it into coefficients in the wavelet domain. The wavelet domain has two key properties advantageous for our image compression/reconstruction system. The multi-decomposition property is the first and the second is parsimony [1].

*2.3.1 Multi-Decomposition.* Because we are using two-dimensional images, we use the two-dimensional wavelet transform. The two-dimensional wavelet transform is a separable transform meaning the transform can be applied in either order [1, 2, 9]. In this research we apply the filter first down the columns and then across the rows. Each transform involves the application of a high pass filter, H, and a low pass filter, L, to create a detailed and coarse approximation to our original signal, respectively. These approximations are decomposed by two to ensure equal number of input samples and wavelet coefficients. Using these two filters which must satisfy certain properties to form a valid wavelet transform, we decompose the standard image, “Lenna” in Figure 2.1, into four separate subbands as seen in Figure 2.2 [1]. The subband names come from the order of filtering. The LH subband means the low pass filter was applied down the columns first and then the high pass filter across the rows. Within each subband, different information is extracted from the image. In the LH subband, the vertical edges are emphasized. The HL subband pulls

out the horizontal edges with the HH subband extracting the diagonal edges. The LL subband is a smoothed version, a coarse approximation, of the original image. Figure 2.3 shows a simple image of the information extracted within each iteration of the two-dimensional wavelet transform.



Figure 2.1: *Original 256 x 256 8-bit gray scale “Lenna” image.*

The wavelet transform continues to decompose the image frame through multiple iterations. The LL subband image, the smoothed image, is again passed through the filters to create four more subbands. These new subbands extract information from the LL subband of the previous iteration and create detailed subbands at a lower scale. Figure 2.4 shows the original “Lenna” image after three iterations of the wavelet transform. The HH, HL, and LH subband of the first iteration are identical to those in Figure 2.2. The first iteration’s LL subband is decomposed through two additional iterations giving two more HH, HL, and LH subbands at coarser scales. The final LL subband is a more smoothed, less detailed, version of the original image because more information has been extracted by each of the other subbands during the iterations. Each of the subbands contains specific information, (horizontal





Figure 2.2: The original image is decomposed into the four subbands. The name of the subband comes from the type of filter and order. *H* is for high pass, and *L* is for low pass. The first filter is applied down the columns, and the second is applied across the rows.

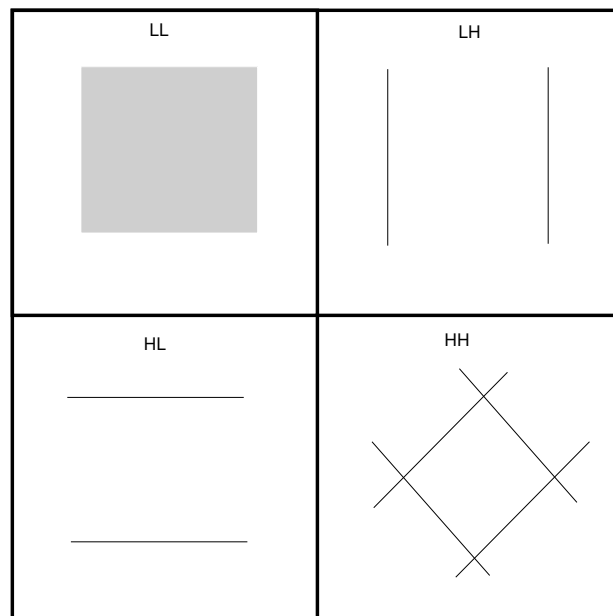


Figure 2.3: A simplified version of the two-dimensional wavelet transform output. The original image is a box which has the vertical edges extracted by the *LH* subband, the horizontal edges extracted by the *HL* subband, and the diagonal edges by the *HH* subband. The *LL* subband contains a smoothed version of the original for information about the edges have been extracted by the other three subbands.

edges, vertical edges, diagonal edges, coarse approximation) at different scales; the property of parsimony allows us to use a small portion of these subband coefficients to reconstruct a quality image.

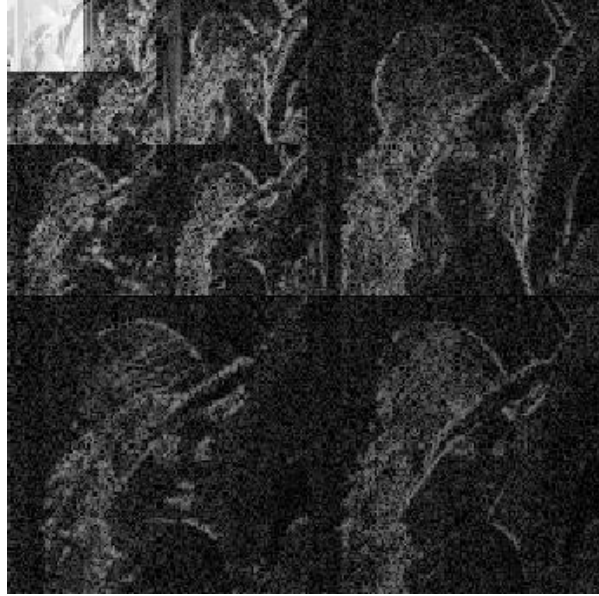


Figure 2.4: The original “Lenna” image after three iterations of the wavelet transform. The first iteration HH, HL, and LH subband are the same as before. The other iterations of the HH, HL, and LH subbands extract more information from the previous LL subband image. The final LL subband is an extremely coarse version of the original image.

*2.3.2 Parsimony.* The property of parsimony means that most of the energy for the image is located in a few significant wavelet coefficients [1,2]. Figure 2.5 shows a bar plot of the logarithmic of the number of coefficients versus their magnitudes. Most coefficients are small. Because the majority of the image energy is located in a few coefficients, we can reconstruct the image with high quality from this small set of coefficients. By including more coefficients with less energy, we do not gain a significant increase in image quality. Figure 2.6 shows the original image, along with five different reconstructions. Each figure uses fewer coefficients to reconstruct the image. When we use only the largest 13.5% of the available coefficients, we still get a high quality image, PSNR > 33dB. Clearly, a small number of wavelet coefficients provide a quality reconstruction. By using a small number of

coefficients, we significantly decrease the number of total bits necessary to transmit the image, creating a more compressed transmission file.

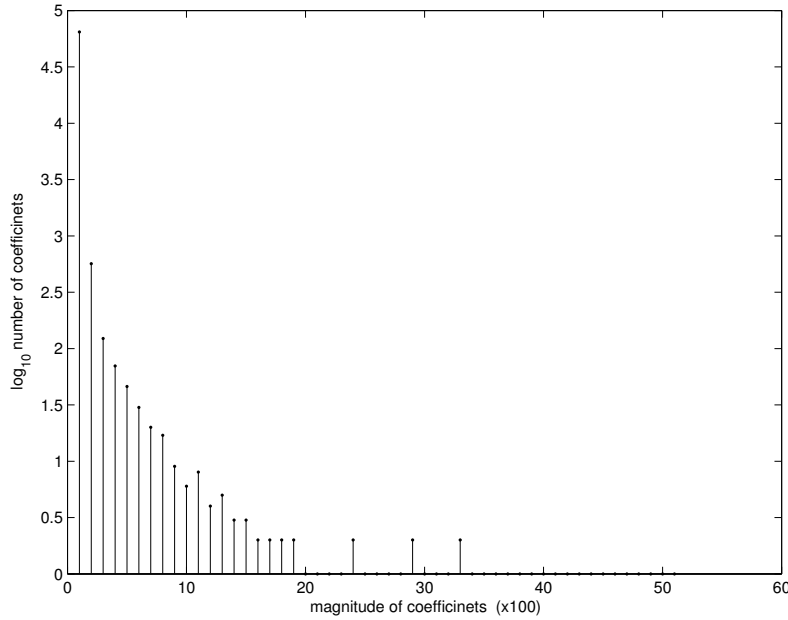


Figure 2.5: *Plot of the logarithmic of the number of wavelet coefficients with the given energy magnitude. Each bar represents a spread of 50 for the coefficients from 0 to the maximum coefficient value, 3,214. Most of the coefficients contain little energy. Most of the energy of the image is contained in a few coefficients.*

## 2.4 Embedding Audio/Video

Current technologies for compressing video with audio typically compress both aspects separately and then transmit the compressed files as two independent packages. This causes potential problems in the reconstruction of the video stream for the audio signal can lose synchronization. Previous research by Zhang and Zheng at Ohio State University addressed this problem by concatenating the audio information to the bottom of the image file [22]. This corrects the synchronization problem in the reconstruction for audio signals are tied to the specific frame. However, this forces the image compression algorithm to operate on an artificial image with statistics vastly different from natural imagery.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 2.6: (a) The original image without any compression. (b) The original image is reconstructed from all 65,536 coefficients giving a PSNR of 51dB. (c) Using the 10,000 largest magnitude coefficients gives a PSNR of 37dB. (d) Using the 8,863 largest magnitude coefficients gives a PSNR of 36dB. (e) Using the 1,000 largest magnitude coefficients gives a PSNR of 25dB. (f) Using the 500 largest magnitude coefficients gives a PSNR of 23dB. Image quality does not noticeably degrade until image (e).

More recent research has used digital watermarking techniques to embed the audio information directly into the image frame [10]. This addresses the audio synchronization problem, and also, permits the file with the embedded information to compress to the same size as the image only compressed file.

## 2.5 Watermarking

Watermarks are information embedded into a source file. Information can be embedded for data hiding, data authentication, medical safety, copy protection, and copyright protection among others [8]. Some of these applications are for secure transmission (data hiding, authentication, and medical safety), while others are for commercial purposes. Guaranteeing illegal copies are not produced (copy protection) is fiscally important to the media outlets. However, the most popular use of watermarking in the digital environment is in copyright protection [4, 17].

Digital watermarks embed information into a digital domain source as opposed to a physical source. The two primary characteristics of digital watermarks are identical to those in the physical world. They are:

- Imperceptibility - This is the property that states embedded information should not distract from the source material. For example if the source is an image, the image quality should remain high after watermarking. If the source is a song, the embedded audio file should not introduce any new pops or hisses.
- Robustness - Also called Security or Strength. The watermark should survive attacks upon it. These attacks come from signal processing for images or audio processing for songs. Because we are working with images, common signal processing attacks include: image compression, filtering, image enhancement techniques, quantization, digital-to-analog conversion, and analog-to-digital conversion [3].

These two characteristics are necessary considerations in the design of a watermarking system. If imperceptibility is not considered, the presence of embedded informa-

tion is noticed which distracts from the quality of the source work. If robustness is not considered, then when the work is compressed for transmission, the watermark can be lost or corrupted.

One secondary characteristic is of importance to us. Extraction without original information allows the system to extract the watermark without any knowledge about the original source work [20]. We can embed the watermark into any source, and the receiver can still extract it without prior knowledge of the unwatermarked source. This is important in video data. We do not wish to send every frame twice, once with the watermark information and once without it, nor do we want the extra overhead of decryption keys. If we are able to extract without original information, we only need to send the video with the watermark. However, before we can transmit the watermarked video, we must quantize and encode the video coefficients.

## 2.6 Quantization/Reconstruction

The quantization process consists of two parts. The two parts are the way the magnitudes of the coefficients are coded into bits and the way the indices and signs of the coefficients are coded. To code the magnitudes we use bit plane coding, while to code the indices and signs we use index coding [15].

*2.6.1 Bit Plane Coding.* Bit plane coding contains two portions: initial bit assignment and bit refinement. The range of the quantization is selected based upon the largest magnitude coefficient,  $C$ , and the number of quantization iterations selected,  $Q$ . During the initial bit assignment, the system checks if the coefficient is greater than or less than  $\frac{C}{2}$ . If the value is larger, it receives a ‘1,’ else a ‘0.’ The range is then broken in half for bit refinement. If the coefficient received a ‘1’ on the first pass, on the second pass, the refinement, we determine if the coefficient is less than or greater than  $\frac{C}{2} + \frac{C/2}{2} = \frac{3}{4}C$  to assign a ‘0’ or ‘1’ respectively. The refinement occurs  $Q - 1$  times. The minus one is because the initial assignment counts as one iteration.

As an example of this coding, let  $C = 128$ ,  $Q = 3$ , and the coefficient = 70. In the initial assignment, we compare 70 to  $\frac{C}{2} = \frac{128}{2} = 64$ . Because 70 is greater than 64, we assign a '1' and decrease  $Q$  to 2. Now we are in bit refinement. We compare 70 against  $\frac{3}{4}C = 96$ . 70 is less than 96, and so we assign a '0' and decrease  $Q$  to 1. After two iterations we have coded 70 as '10.' For the third pass, we compare 70 to  $\frac{5}{8}C = 80$ . 70 is less than this value, so again we assign a '0' and decrease  $Q$ . Because  $Q$  now equals 0, we have concluded our iterations of coding. We have coded 70 as '100.' Now we need to reconstruct the coefficient.

Reconstruction is recreating the quantized coefficient from the bit code. Because we know the largest magnitude coefficient,  $C$ , and the bit code, we can reconstruct the coefficient. With our example above, we had  $C = 128$  and bit code = '100.' Taking the first bit, '1,' we know the coefficient is between  $\frac{C}{2} = 64$  and  $C = 128$ . The second bit, '0,' tells us the coefficient is less than  $\frac{3}{4}C = 96$ . We already know the coefficient is greater than 64. We are shrinking the unknown range for the coefficient. The third bit, '0,' tell us the coefficient is also less than  $\frac{5}{8}C = 80$ . Therefore, we now know the coefficient lies somewhere between 64 and 80. We reconstruct the coefficient as the midpoint value in this region. The reconstructed value for 70 in this example is  $\frac{64+80}{2} = 72$ . By increasing the number of iterations, increasing  $Q$ , we will, on average, decrease the difference between the reconstructed and original values.

*2.6.2 Index Coding.* The bit plane coding converts the magnitudes into bits while the index coding converts the index of the coefficient, the location, into bits. The index coding also takes into consideration the sign of the coefficient which was not considered previously.

Because of the multi-decomposition property of wavelets as seen in Figure 2.2, we process the coefficients in the order specified in Figure 2.7. We start with the coarse region. Moving to the first LH region, we process vertically through the coefficients. This maintains the vertical nature of this subband. Next we process the

first HL region by taking the coefficients horizontally. By maintaining the order of each subband, we are able to minimize the distance between significant coefficients and exploit correlations in the wavelet domain [12].

1	2	5	8	17	24	25	32
3	4	6	7	18	23	26	31
9	10	13	14	19	22	27	30
12	11	15	16	20	21	28	29
33	34	35	36	49	50	51	52
40	39	38	37	53	54	55	56
41	42	43	44	57	58	59	60
48	47	46	45	61	62	63	64

Figure 2.7: *Order coefficients will be processed to minimize the distance between significant coefficients. This follows the information extracted within each subband. For the HL subband, we process horizontally, while for the LH subband we process vertically.*

When we code the indices, we use the first difference. We order the significant coefficient indices in increasing order. This guarantees that the current index is always greater than the previous index. The first difference stores a value that is relative to the index stored before it. For example, if we have a list of indices  $I=[1\ 4\ 7\ 19\ 22]$ , the stored values are  $I'=[1\ 3\ 3\ 12\ 3]$ . The difference between the index and its predecessor is stored. To reconstruct the indices, we sum them together. Given



$I'$  as before, we would get  $[1 \ 1+3 \ 1+3+3 \ 1+3+3+12 \ 1+3+3+12+3]$  giving  $[1 \ 4 \ 7 \ 19 \ 22]$  which is our original  $I$  value.

When we coded the magnitudes, we did not take into account the sign. In the index coding we include the sign. We attach the sign, '+' or '-', to the first difference of the indices. Given our indices  $I=[1 \ 4 \ 7 \ 19 \ 22]$  as before, and the corresponding coefficients,  $[42 \ -33 \ 19 \ 12 \ -8]$ , we would get the first difference  $I'=[+1 \ -3 \ +3 \ +12 \ -3]$ .

We take this first difference and convert to binary.  $I'$  in binary is  $[+0001 \ -0011 \ +0011 \ +1100 \ -0011]$ . To save bits, we remove the leading '0's as they offer no information. This gives  $[+1 \ -11 \ +11 \ +1100 \ -11]$ . We now see that each set of bits starts with a '1.' Removing these initial '1's, we get the final output index code  $IC=[+ \ -1 \ +1 \ +100 \ -1]$ . In this example for the four indices with their associated signs, we need only code the four signs plus six bits. This is a significant savings as initially we had the four signs in addition to 16 bits. These coding techniques of [15] are used in the Mendenhall Digital Watermarking System [10].

## 2.7 Mendenhall's Digital Watermarking System

In 2001 Mendenhall created a digital watermarking system that embedded the audio for a video stream into the image frames for transmission [10]. The system entails embedding the audio bits into the image frame using a digital watermark, quantizing the post-watermarked image coefficients, transmitting across a lossless channel, reconstructing the image coefficients, and extracting the audio bits from the reconstructed image coefficients as seen in Figure 2.8. The system uses a constant embedding strength and a sufficient number of quantization iterations to guarantee perfect audio bit extraction.

The stereo audio channels are combined into one bit stream. The bytes are combined such that they alternate between a right channel byte and left channel byte. This stores bytes of audio that are heard at the same time physically in a stream close to one another. The audio bit stream is divided into blocks of bits for

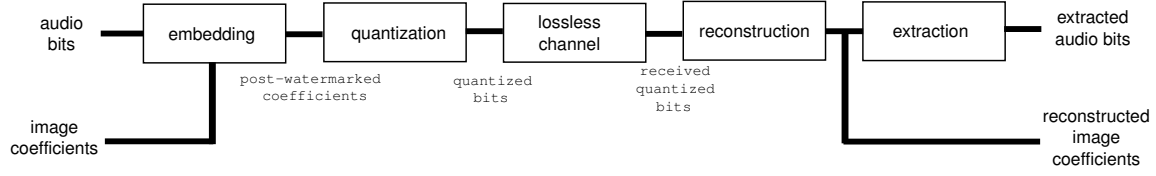


Figure 2.8: *The five sections of Mendenhall's Digital Watermarking System embedding an audio bit stream into the image frame.*

each image frame. If there are five frames and 20,000 audio bits, then each frame will contain 4,000 audio bits.

These audio bits are embedded into the image coefficients using the digital watermarking technique described in [16] based upon work by [18]. The post-watermarked coefficient is based upon the coefficient value, the embedding strength, and the audio bit. Using modulo arithmetic (explained in Equation 2.2), the coefficient is dropped to the closest multiple of the embedding strength,  $S$ .

$$\text{modulo}(a, b) = a - b * \text{floor}\left(\frac{a}{b}\right) \quad \text{for } b \neq 0 \quad (2.2)$$

If the coefficient is greater than zero, subtraction is used, or if less than zero, addition. If the audio bit is a '1,' then  $\frac{S}{4}$  is added to this altered coefficient for positive coefficients and subtracted from for negative coefficient to create the post-watermarked coefficient. To embed a '0,'  $\frac{3}{4}S$  is used. This forces every post-watermarked coefficient to be a multiple of the embedding strength,  $S$ , plus  $\frac{S}{4}$  if containing an embedded '1' and  $\frac{3}{4}S$  if containing a '0.'

Using the bit plane and index coding as explained previously, these post-watermarked coefficients are converted to bits for transmission. The system assumes a lossless channel for transmission such that every bit sent is received without error. Upon receiving the transmitted bits, they are reconstructed as explained earlier to get the reconstructed image coefficients.

The extraction process uses the reconstructed image coefficients and the known embedding strength to recreate the audio stream. Again using the technique de-

scribed in [16], the system uses modulo arithmetic to determine the embedded bit value. The system checks if *modulo*(reconstructed coefficient,  $S$ ) is greater or less than  $\frac{S}{2}$  where  $S$  is the embedding strength. If the value is greater than or equal to  $\frac{S}{2}$ , the the system determines a ‘0’ was embedded into the coefficient. If less than, the system determines a ‘1’ was embedded. The extracted bit stream is reconstructed into stereo audio.

The reconstructed image coefficients still contain the embedded audio bits, but Mendenhall used an embedding strength and number of quantization iterations to ensure the reconstructed image maintained a high PSNR as compared to the original. The embedding strength,  $S$ , and number of quantization iterations,  $Q$ , were also set to guarantee perfect extraction of the audio bits. In this research, we allow for the possibility of incorrectly decoding audio bits. Such errors can be addressed by error correcting codes.

## 2.8 Bose-Chadhuri-Hocquenghem Error Correcting Code

Binary error correction codes (ECC) not only detect bit errors but are able to correct some. The Bose-Chadhuri-Hocquenghem (BCH) code is a type of cyclic code that is a subset of linear block codes [13]. Cyclic codes are characterized by two parameters,  $n$  and  $k$  with  $n > k$ .  $k$  bits of information are encoded into an  $n$ -bit codeword. For each  $k$  bits of information there exists only one  $n$ -bit codeword that it will be encoded as. These codeword bits are transmitted instead of the actual information bits. The decoder checks the received bit codeword and compares it against a look-up table of known bit codewords for the  $n, k$  parameters. If the received codeword is in the look-up table, the  $k$  information bit word is returned. If the codeword is not in the look-up table, the decoder must determine what the information word should be. Because there are more codewords than information words,  $2^n > 2^k$ , the decoder uses the Hamming distances between the codewords to determine the information word. The Hamming distance is the amount the codewords differ from one another [13]. For example, two codewords of [1 0 0 1] and [1

0 1 0] have a Hamming distance of two for they differ only in the last two bits. The greater the distance, the more errors the code can correct. As errors are introduced, the codewords become altered from their original values. If the codewords differ enough to be a different codeword, the codeword is decoded incorrectly. By having a large Hamming distance, the likelihood a codeword is decoded incorrectly is less.

The BCH code does not guarantee perfect correcting ability. The tradeoff between  $n$  and  $k$  determines the number of bits in the codeword that can be corrected. The number of bits that can be corrected is given by  $t$ , which varies with the choice of  $n$  and  $k$ . As  $k$  decreases for a constant  $n$ ,  $t$  increases. This means, as we increase the overhead to encode the information words, we gain more error correction.

## 2.9 Summary

This chapter discussed how the HVS is an imperfect system. As a result, a reconstructed image does not need to be identical to the original for the human eye to perceive no error. This allows for information embedding and lossy image compression. PSNR is the industry standard used to measure image quality which we also use.

We exploit two important properties of wavelets in our system. Because of the multi-decomposition property of wavelets, we are able to process the wavelet coefficients of an image by separate bands. The parsimonious property of wavelets allow us to retain a small number of the possible coefficients to reconstruct a quality image.

Previous methods have been tried for audio/video transmission with synchronization problems or transmission size being the bottlenecks. Using digital watermarks, Mendenhall was able to embed the audio information into the image frames. Bit plane coding and index coding of the post-watermarked wavelet coefficients of the image frame gave a way to transmit the information and reconstruct it. Mendenhall used a constant embedding strength and a high number of quantization iterations

to guarantee perfect extraction of the audio bits and a high quality reconstructed image. Using this system along with a Bose-Chadhuri-Hocquenghem (BCH) error correction code, we are now able to introduce and explain tradeoffs to meet the user's needs in the audio/video compression and transmission system.

### III. Methodology and Design

This chapter includes the specific approach to analyze and modify the information watermarking and wavelet-based video compression developed by Mendenhall [10]. It explains the two channel structure of our system. The variables that dictate the system's state and the three measurements used to character the state are described. The chapter concludes by discussing the quantization process and the Information Bit Error Rate (IBER) plot regions which demonstrate some restrictions on the variables.

#### 3.1 Two Channel Structure

The entire watermarking and compression system can be categorized as a two channel system as seen in Figure 3.1. The video bits transmitted are sent across a lossless channel. No errors are introduced into the bit stream by the lossless channel.

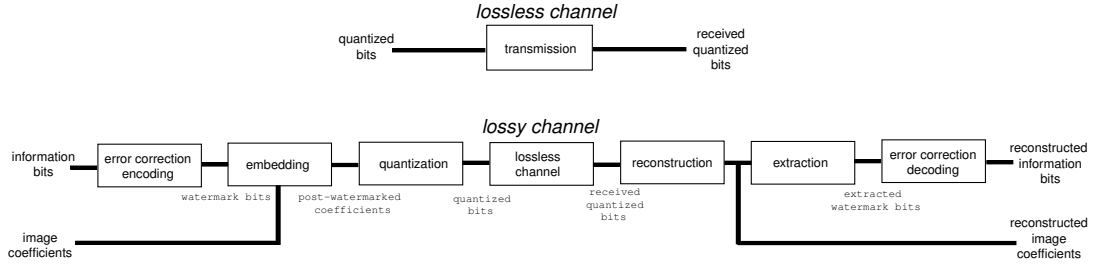


Figure 3.1: Two Channel representation of watermarking and compression system.

The information bits and image coefficients are sent across a lossy channel. Upon reconstruction, the transmitted quantized bits are combined to give the reconstructed image coefficients. These reconstructed image coefficients differ from the original image coefficients due to the information bits embedded into them in addition to quantization.

The information bits are encoded and then embedded into the image coefficients which are quantized before transmitting across the lossless channel. The

quantization is considered an attack on the embedded watermarked bits since quantization alters the reconstructed image from its original value. This potentially causes bit loss or error during the bit extraction process for we no longer have an exact copy of the post-watermarked coefficient before quantization. As explained in Section 2.8, the decoding should correct some of these errors, but it cannot correct them all and may in fact introduce some of its own decoding errors into the final reconstructed information bits. If the errors introduced exceed the Hamming distance of the codeword, the errored codeword may be decoded incorrectly. Instead of recognizing it as a errored copy of the original codeword, it may be seen as an errored copy of a different codeword. This is why the ECC costs. We need a Hamming distance greater than the amount of introduced error.

### 3.2 Variables

The following variables dictate the state of the system:

- $N$  is the number of significant wavelet coefficients from the current image frame.  $N$  is clearly less than or equal to the total number of pixels in the image. We want  $N$  large enough to achieve a quality image. A quality image is similar enough to the original image that the human visual system cannot differentiate between the two. Also, as  $N$  increases, more watermark bits can be incorporated. This is a one to one ratio; one and only one watermark bit can be embedded into each coefficient. Therefore,  $N$  is also the maximum number of watermark bits. More watermark bits allow for increased information transmission or the incorporation of additional error correcting capability.

However, we want  $N$  small enough to decrease the bit to pixel ratio from a compression standpoint. The wavelet property of parsimony states that most of the image's energy is in a small number of significant coefficients. By increasing  $N$ , we use more of the non-significant coefficients. Keeping these non-significant coefficients requires a finer degree of quantization. These non-

significant coefficients are much smaller than the significant ones and require additional refinement passes in the quantization process. Quantizing to a finer degree means we will store more bits, opposing the compression desire: to limit the amount of bits for transmission.

- $T$  is the number of bits needed to fully define the maximum coefficient of the image as in Equation 3.1 where  $w_i$  are the wavelet coefficients of the image. Thus the largest coefficient always has magnitude less than  $2^{(T+1)}$ .  $T$  is commonly on the order of 11 or 12.

$$T = \text{floor}(\log_2(\max(\text{abs}(w_i)))), \quad (3.1)$$

This variable is image dependant. The value of  $T$  changes based upon the current image.

- $Q$  is the number of quantization iterations. This is related to how many bits are stored for transmission. For a given  $Q$ , the quantization levels go from  $2^T$  to  $2^{(T-Q+1)}$ . Therefore, the smallest watermarked coefficient the quantization step can store is  $2^{(T-Q+1)}$ . As  $Q$  increases, the reconstructed coefficients are more accurate which increases the image quality. Also with detailed reconstructed coefficients, the difference in a coefficient with an embedded '1' and the same coefficient with an embedded '0' is noticeable and therefore extractable.  $Q$  also impacts the number of transmission bits. Increasing  $Q$  increases the refinement of the coefficients which increases the number of bits required to describe each coefficient. Therefore, we wish to keep the number of quantization iterations large enough to give sufficient detail within the coefficients and improve image quality but small enough to keep the number of bits to transmit low.
- $S$  is the strength (robustness) of the watermark. This determines how much of an attack the watermark can survive. The attack on the watermark in this system comes from the quantization of the watermarked coefficients. The larger the  $S$ , the more robust the watermark is to attacks. The difference between



an embedded ‘1’ and an embedded ‘0’ is  $\frac{S}{2}$ . By increasing  $S$ , we increase the separation between each embedded watermark value. By increasing the separation, we can compensate for attacks that corrupt the value of the post-watermarked coefficient. If the post-watermarked coefficient is altered less than  $\pm\frac{S}{4}$ , the watermarked bit is extracted without error. This implies we want a large  $S$ .

However, we also have reasons to keep  $S$  small. By embedding a watermark bit into the coefficient, we modify that coefficient. With a large  $S$ , we corrupt the coefficient by a large amount. The PSNR of the reconstructed image is based on the coefficient being similar to the pre-watermarked coefficient. To maintain image quality, we want a small  $S$  so that we do not significantly distort the wavelet coefficients. Another reason to keep  $S$  small is for quantization. By increasing the level of coefficient distortion, we may create a post-watermarked coefficient too small for our quantization level. Our quantization has a lower limit; the smallest value it can quantify. If we distort the coefficient too much, such that it drops below this minimum value, then a coefficient which was significant is ignored.

- $t_o$  is the smallest level of quantization. This implies  $t_o$  is the value of the smallest coefficient that can be quantified. The minimum value is independent of  $S$ . The smallest wavelet coefficient after watermarking must be greater than this value.

$$t_o = 2^{(T-Q+1)}. \quad (3.2)$$

- $N_Q$  is the largest  $N$  for a specific  $Q$  value and specified image file. Sorting the wavelet coefficients of the image by magnitude,  $N_Q$  is the maximum number of coefficients starting with the most significant and working down through the non-significant coefficients that can be quantized for the given image and number of quantization iterations,  $Q$ . The coefficient just beyond the  $N_Q$ th coefficient is too small to be quantized. It will be skipped during quantization.

Because of this, we keep the number of coefficients we use,  $N$ , less than or equal to  $N_Q$ , never greater. As shown in Equation 3.3, to use a coefficient, it must be quantifiable and therefore, must be one of the first  $N_Q$  significant coefficients.

$$N \leq N_Q \quad (3.3)$$

- $M$  is the number of information bits encoded in the image frame. This value is always less than or equal to the total number of watermark bits and clearly, the number of significant wavelet coefficients,  $N$ .

$$M \leq N. \quad (3.4)$$

- $k$  is the number of information bits for an  $n, k$  Bose-Chahuri-Hocquenghem (BCH) error correcting code.
- $n$  is the length of the codeword the  $k$  information bits will be mapped into for an  $n, k$  BCH error correcting code;  $n > k$ .
- $t$  is the number of bits out of  $n$  that can be corrected in an  $n, k$  BCH error correcting code. Given an  $n$  and  $k$ ,  $t$  is specified. An  $n, k$  code can correct error rates less than  $\frac{t}{n}$ .

### 3.3 Measurements

Three fundamental issues are relevant to the state of the system. They are the image quality, the quality of the transmitted information bits, and the rate of transmission. PSNR is used to measure the image quality, while IBER measures the quality of the transmitted information. Bit rate is the third measure used to quantify the rate of transmission. If the image is of poor quality, the system has created a bad but still useable result. Likewise, a large transmission file will simply take longer to send but will not ruin the system. However, if the transmitted information bits are unusable, the system has failed.

*3.3.1 Peak Signal to Noise Ratio.* As explained in Section 2.2, the Peak Signal to Noise Ratio (PSNR) is the quantitative measurement for image quality commonly used by the image processing community. Thus, we also use the PSNR to measure image quality for this system.

*3.3.2 Information Bit Error Rate (IBER).* The Information Bit Error Rate (IBER) is calculated as the number of information bits extracted incorrectly divided by the total number of information bits transmitted,  $M$ .

An IBER of 50% is the worst case. Because the information bits are either a ‘1’ or a ‘0,’ guessing at the extracted information bit stream is probabilistically just as accurate as the system. If  $S$  is large, this occurs when the quantization level is not fine enough to quantify all the watermarked coefficients. In this situation, one or more of the coefficients will be skipped in quantization causing a *Dropped Bit Error*. Therefore upon extraction, this dropped bit is not discovered by the extractor. The extractor’s output contains portions of the original information bits shifted because of the dropped bit. Since the bits are shifted, the extracted information bits contain 50% error.

If  $S$  is small, an IBER greater than 0% means that there are no dropped bits, but a problem extracting the information bits exists called *Extraction Error*. This occurs because the difference between an embedded ‘1’ and ‘0’ is not great enough. This can be fixed by either increasing the quantization iterations,  $Q$ , to a finer degree or by increasing the watermark strength,  $S$ , which increases the separation between an embedded ‘1’ and an embedded ‘0.’

*3.3.3 Bit Rate.* The bit rate is the ratio of the number of transmitted bits to the number of pixels. This gives the number of bits necessary to represent each pixel. No compression would be a bit rate of eight because we are using eight bits for each of our  $256^2$  pixels. The smaller the rate, the more compression.

### 3.4 Quantization

As described in Section 3.2,  $Q$  dictates the number of quantization iterations to perform. In Mendenhall's system, the number of quantization iterations was based on achieving perfect extraction of the information bits without any error correction code [10], IBER=0%. To this end, the  $Q$  value was calculated based on the minimum coefficient after the watermarking process. The relationship is explained in Equation 3.5 with  $T$  defined in Equation 3.1.

$$\text{minimum post-watermarked coefficient} = 2^{(T-Q+1)}. \quad (3.5)$$

For this thesis investigation, we are not restricted to this value for  $Q$ . Instead, because we are making the number of iterations variable, we specifically choose a  $Q$  value less than that specified by Equation 3.5.

After specifying the value of  $Q$ , we now know the minimum post-watermarked coefficient,  $t_o$ , that we can quantify defined in Equation 3.2. This equation takes into account the  $T$  from our image and the  $Q$  we have chosen. Any value less than  $t_o$  is skipped during quantization and so never seen during extraction. The quantization and reconstruction bins are the same. They are  $\frac{t_o}{2}$  wide. Figure 3.2 shows for a given  $T = 11$  and  $Q = 8$ , the distribution of the bins. The minimum value is at  $t_o$  with every other bin edge at  $j * t_o$  where  $j$  is the bin number. For this example,  $t_o = 16$ .

The reconstruction values are created from the quantization bit output that is transmitted across the lossless channel. These values lie halfway between each bin. Therefore, they lie at

$$\text{reconstruction values}_j = t_o + j * \frac{t_o}{2} + \frac{t_o}{4}. \quad (3.6)$$

Figure 3.3 shows the reconstruction values laying between the bins. The reconstruction values are the short lines, while the bin edges are the tall lines.

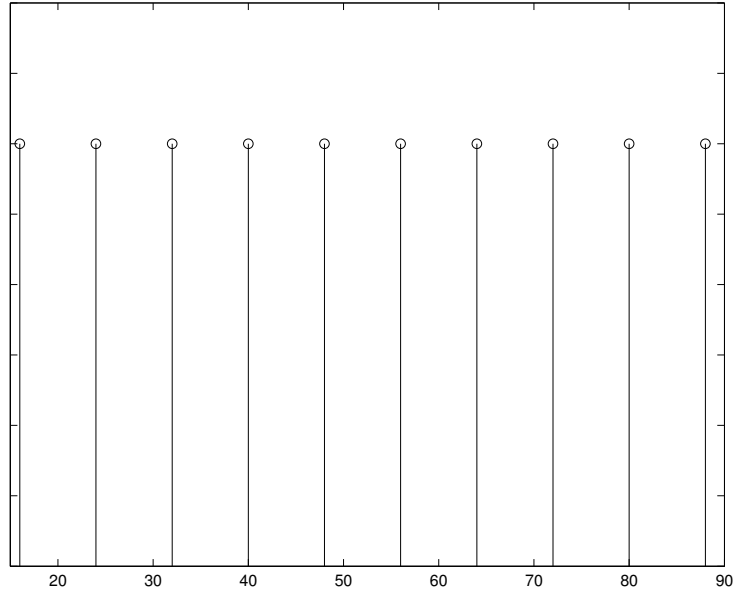


Figure 3.2: *Bin layout based on  $T = 11$  and  $Q = 8$ .*

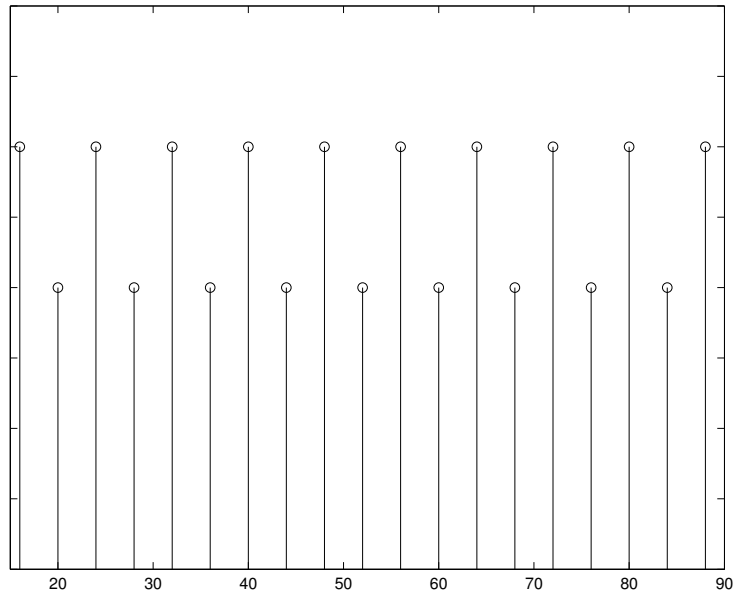


Figure 3.3: *Bins with reconstruction values for  $T = 11$  and  $Q = 8$ . The short lines are the reconstruction values, while the tall lines are the bins.*

### 3.5 Regions on the IBER plot

Figure 3.4 shows experimentally Information Bit Error Rates (IBER) for a constant  $Q$  and  $T$  while varying  $S$ . The three regions are the *Extraction Error* region, *Perfect Extraction* region, and the *Dropped Bits* region.

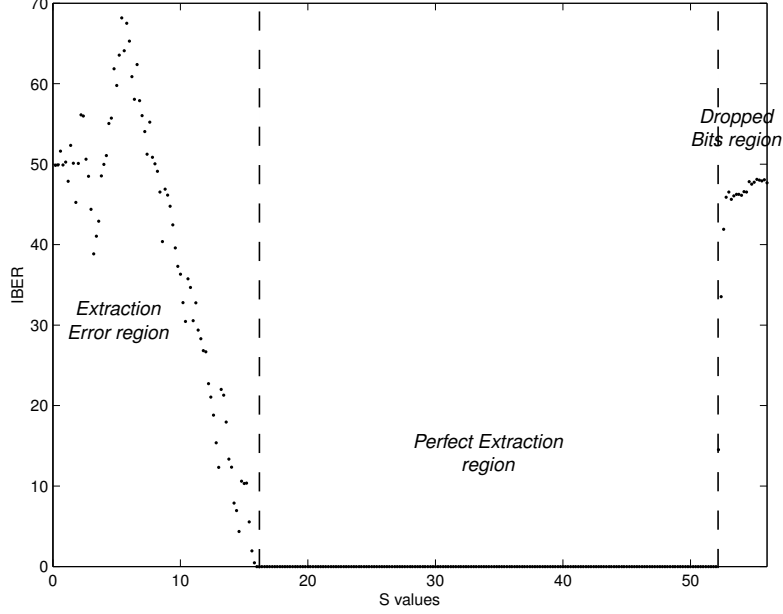


Figure 3.4: *Three Regions of Information Bit Error Rate versus  $S$  using experimental results.*

**3.5.1 Dropped Bits region.** The *Dropped Bits* region is defined by an IBER of 50% with a large  $S$ . The  $S$  in this region is too strong, large, for the system, pushing the post-watermarked coefficient below the minimum quantization level,  $t_o$ . When embedding a ‘1’ or ‘0,’ the coefficient is altered by our watermarking method explained in section 2.7 which uses modulo arithmetic to drop the coefficient to a multiple of  $S$ . If  $S$  is too large, dropping to the next lowest multiple of it may drop the post-watermarked coefficient below  $t_o$ . During quantization, this post-watermarked coefficient will be missed. Because the value is smaller than the minimum quantizable value, it will not be quantized to any value. It is skipped completely. Upon reconstruction, the system cannot reconstruct this coefficient because it was never quantized. During extraction, the bit in this coefficient is not extracted. Appearing

to have dropped a bit, the extracted information bit stream is shifted by one where the bit was dropped causing a 50% IBER.

*3.5.2 Extraction Error region.* The *Extraction Error* region is defined by an  $S$  less than the *Perfect Extraction* region. Within this region, the IBER is not always 50% like in the *Dropped Bit* region. Instead, the IBER takes on values from 0% to 50% because the  $S$  is not large enough to push an embedded ‘1’ and an embedded ‘0’ into different quantization/reconstruction bins. As mentioned before, the difference between an embedded ‘1’ and a ‘0’ is  $\frac{S}{2}$  giving the difference from one embedded ‘1’ to the next embedded ‘1’ as  $2 * \frac{S}{2} = S$ . However, the width of the reconstruction bins is based upon  $T$  and  $Q$ , independent of  $S$ . The bins are  $\frac{t_o}{2}$  wide with  $t_o$  defined in Equation 3.2. When the difference between an embedded ‘1’ and the next embedded ‘1’ becomes less than the width of two bins,  $2 * \frac{S}{2} = S < 2 * \frac{t_o}{2} = t_o$ , more than just two embedded bits will be within the range of the two bins. This causes extraction error, for each bin can only return one extracted value, a ‘1’ or a ‘0,’ never both. The IBER increases while the  $S$  decreases because more bins contain multiple embedded bits. When each bin contains one embedded ‘1’ and one embedded ‘0,’ we get 50% IBER. Only half the bits are extracted correctly.

*3.5.3 Perfect Extraction region.* In this region, the  $S$ ,  $Q$ , and  $T$  mesh perfectly. All the post-watermarked coefficients are quantized such that only one embedded value lies within the quantization/reconstruction bin. During the watermarking step, a coefficient will be placed within different quantization/reconstruction bins based upon whether a ‘1’ or a ‘0’ is embedded within it. Therefore, all coefficients within a reconstruction bin contain embedded ‘1’s or they all contain embedded ‘0’s. Upon reconstruction, all the embedded bits are extracted perfectly.

*3.5.4 Theoretical Result.* Figure 3.5 shows given the  $Q$  and image file (which gives  $T$ ), the IBER can be predicted for the *Extraction Error* region. This is the region we wish to operate in. We cannot operate in the *Dropped Bits* region

because we cannot operate with 50% IBER. The *Perfect Extraction* region gives many values of  $S$  to choose, but only one value makes sense. This value is  $S = t_o$ . Any  $S$  larger than this in the region gives the same IBER but decreases the PSNR of the image because the coefficients are being altered more from their original values. We now have an upper limit on  $S$ ,  $t_o$ . The lower limit on  $S$  is where IBER first hits 50% with decreasing  $S$ . This occurs at  $S = \frac{t_o}{2}$ . At this  $S$ , two embedded values within each quantization/reconstruction bin first occur. Less than this  $S$ , every bin contains two or more embedded values. Increasing the number of embedded values beyond two does not change the probability of that bin causing an error. Whether the bin contains one or more than one embedded values makes the bin a good bin for quantization/reconstruction or a bad bin. Therefore, we have a range for useful  $S$ :

$$\frac{t_o}{2} < S \leq t_o. \quad (3.7)$$

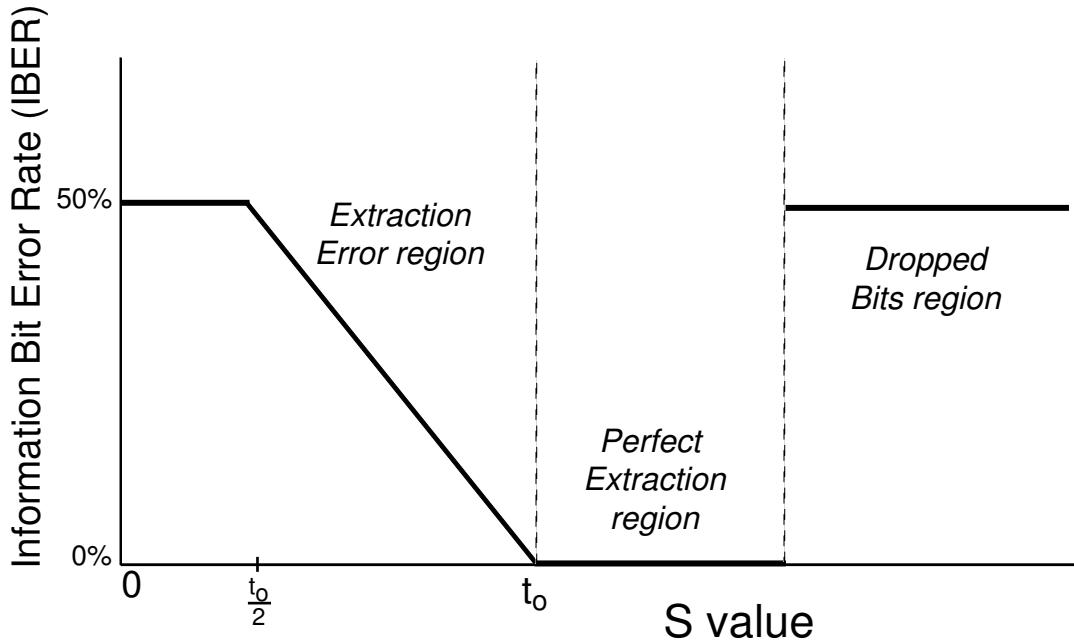


Figure 3.5: *Three Regions of Information Bit Error Rate versus  $S$  using theoretical calculations.*



### 3.6 Summary

This chapter explained the two channel design of the system. The lossless channel implies that the bits we send are received without any error. The second channel is lossy. This channel introduces distortion and error into the image coefficients and information bits, respectively. The variables that define the system parameters were also explained. Some of the interrelationships between these variables will be explained further in Chapter IV. These interrelationships achieve the goals specified by the user. Quantization was further explained in the context of how the system actually operates. Finally, we explained the *Extraction Error* region on the IBER versus  $S$  plot of Figure 3.5 where we operate.

## IV. Results and Analysis

This chapter explains the results of our analysis. It answers the question: given a requirement by the user, what parameters should the system use? The measurements are the information bit error rate (IBER), compression size based on the number of wavelet coefficients and quantization level, bit rate, IBER with error correction code (ECC), and PSNR.

### 4.1 Given IBER acceptable to the user, $T$ , and $Q$ , choose $S$

One of the uses of this system is to specify an IBER, an image, and a compression size specified by  $Q$ . Using this information we can specify the minimal  $S$  to achieve these goals.

As explained earlier in Section 3.5,  $Q$  and  $T$  dictate the minimum coefficient that can be quantized,  $t_o$ . They also dictate the width of the quantization/reconstruction bins which is  $\frac{t_o}{2}$ , and thus the reconstruction values.  $S$  dictates the distance between embedded bits. Between an embedded ‘1’ and embedded ‘0’ is  $\frac{S}{2}$ . Between an embedded ‘1’ and the next embedded ‘1’ is  $2 * \frac{S}{2} = S$ . When  $S = t_o$ , we get exactly one embedded bit value within each bin because the width of the bins and the distance between two embedded bits are equal. This choice of  $S$  is at the lower end of the *Perfect Extraction* region. The embedded values and the bin values all occur at multiples of  $t_o$ . Equation 4.1 shows where the embedded values occur.

$$\frac{S}{4} + j\frac{S}{2} \quad \text{for all } j \text{ values} \quad (4.1)$$

For even  $j$ , these are embedded ‘1’s while for odd  $j$ , these are embedded ‘0’ values. Figure 4.1 shows how only one embedded value lies within each bin. The tall lines are the bin edges, the middle lines are the reconstruction values, while the short triangles are the embedded ‘1’s, and the short circles are the embedded ‘0’s for  $T = 11$ ,  $Q = 8$ , and  $S = t_o$ .

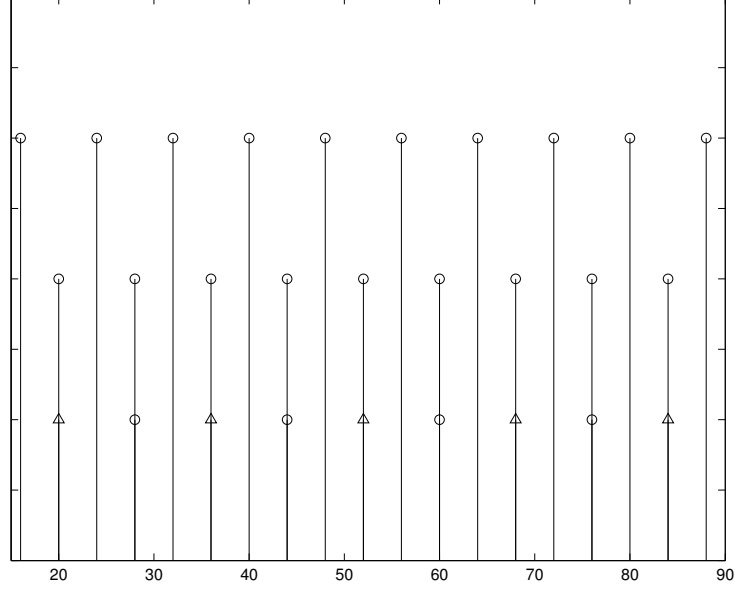


Figure 4.1: Bin edges, reconstruction values, and embedded bit values. Tall lines are the bin edges, the middle lines are the reconstructed vales. The short triangles are the embedded ‘1’s, and the short circles are the embedded ‘0’s.

As  $S$  decreases, we now enter the *Extraction Error* region. Because  $S < t_o$ , some bins contain both an embedded ‘1’ and an embedded ‘0.’ These are the bad bins that cause extraction error. The extraction error occurs because each bin contains only one reconstruction value, the middle line from Figure 4.1. The system can only extract one bit value from the reconstruction value as explained in Section 2.7. Therefore, if an embedded ‘1’ and ‘0’ are both within the same bin, both post-watermarked coefficients are quantized and reconstructed as the same value. The system extracts the same bit from these coefficients because they are reconstructed as the same value. The extracted bit is correct for only one of the embedded bits. As  $S$  decreases, the difference between an embedded ‘1’ and embedded ‘0’ shrinks compared to the width of the bins which is independent of  $S$ . More bins become bad bins. When  $S = \frac{t_o}{2}$ , an embedded ‘1’ and an embedded ‘0’ are within every bin. This causes the IBER to be 50%. Figure 4.2 shows the layout of the embedded bits with respect to the bin values for  $S = \frac{3}{4}t_o$ . Figure 4.3 shows how each bin contains both an embedded ‘1’ and ‘0’ when  $S = \frac{t_o}{2}$ .

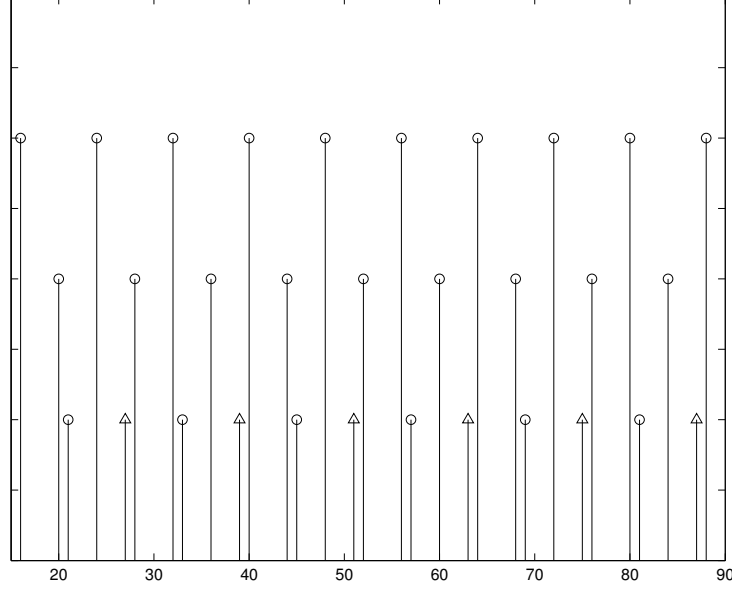


Figure 4.2: Bin edges, reconstruction values, and embedded bits values for  $S = \frac{3}{4}t_o$ . Tall lines are the bin edges, the middle lines are the reconstructed values. The short triangles are the embedded '1's and the short circles are the embedded '0's. A third of the bins contain both an embedded '1' and embedded '0.' These are the bad bins that cause the extraction error.

As seen in Figure 3.5 previously, we can predict the IBER for a given  $T$  and  $Q$  for varying  $S$  in the *Extraction Error* region. Our experimental results in Figure 3.4 prove that our prediction of a linearly decreasing line in this region is correct. When  $S = \frac{t_o}{2}$ , we get an IBER=50%. This drops to 0% when  $S = t_o$ . IBER depends upon the relationship between  $S$  and  $t_o$ . Equation 4.2 shows the IBER function within the *Extraction Error* region as this is the only region of interest.

$$IBER = -\frac{50}{\frac{t_o}{2}}S + 100 \quad \frac{t_o}{2} \leq S \leq t_o \quad (4.2)$$

This is the only region of interest for  $S$  values larger than  $t_o$  does not decrease the IBER any for it already is at 0%. For  $S$  values less than  $\frac{t_o}{2}$ , we get 50% which is uncorrectable. At these values of  $S$ , half the bits are wrong. The reconstructed bit stream is useless. This is why we restrict the values of  $S$  to be between  $\frac{t_o}{2}$  and  $t_o$ . Therefore, for a given IBER,  $T$ , and  $Q$ , we can find the smallest  $S$  to meet the IBER requirement using Equation 4.2.

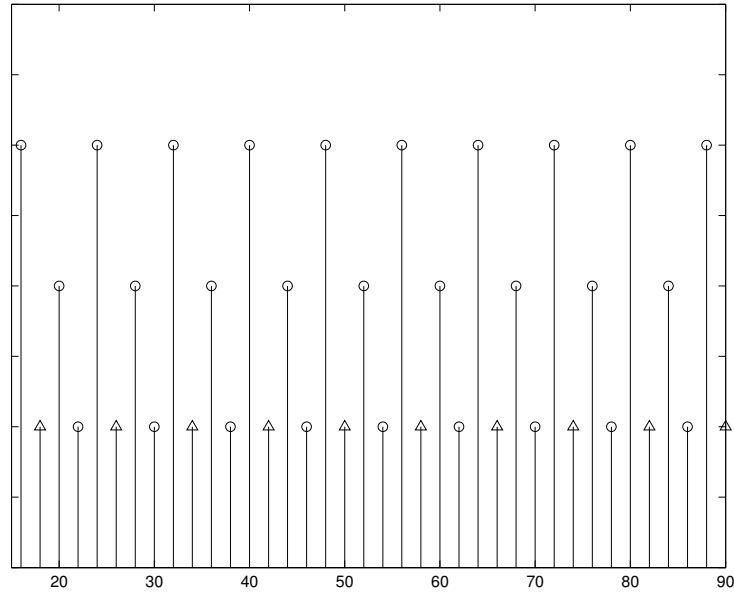


Figure 4.3: Bin edges, reconstruction values, and embedded bits values for  $S = \frac{t_o}{2}$ . Tall lines are the bin edges, the middle lines are the reconstructed values. The short triangles are the embedded '1's and the short circles are the embedded '0's. Every bin contains both an embedded '1' and embedded '0.' Because the system can only extract one of these embedded bit values from each reconstructed values, half of the extracted bits will be extracted wrong giving an IBER=50%.

#### 4.2 Given $T$ , $Q$ , and $S$ , choose $N_Q$

$N_Q$  as explained in Section 3.2 is the largest number of coefficients from the given image that can be used for watermarking and still be quantized without introducing errors due to dropped bits. As shown earlier, the amount of watermark bits equals the number of wavelet coefficients retained, so  $N_Q$  also impacts the amount of information bits we can send. The number of coefficients retained along with the quantization level,  $Q$ , directly affects the amount of bits we transmit which dictates the bit rate. The prediction of  $N_Q$  clearly impacts many requirements.

To predict  $N_Q$ , the image coefficients are sorted by magnitude with the largest being at  $N = 0$  and decreasing in magnitude as  $N$  increases.  $N_Q$  is based on the  $T$  from the image frame, the  $Q$  for the quantization limit, and the  $S$  which tells how much each coefficient will be altered. For a given set of parameters,  $T$ ,  $Q$ , and  $S$ , and a given image, we get different  $N_Q$  values. The  $N_Q$  value can change significantly when selecting different parameters with the same image because of the modulo arithmetic used in calculating the post-watermarked coefficients as seen in Equation 4.3 with *modulo* explained in Equation 2.2.

$$\begin{aligned} \text{post-watermarked coefficient} &= \text{coefficient} - \text{modulo}(\text{coefficient}, S) + \left(\frac{S}{4}, \frac{3}{4}S\right) \\ &\text{based on whether embedding a '1' or a '0' with coefficient} \geq 0, \end{aligned} \tag{4.3}$$

The smallest post-watermarked coefficient within the range of  $N_Q$  must be large enough to still be quantized by the given  $Q$ . Therefore, this value must be larger than  $t_o$ , the smallest quantizeable value. For coefficients less than zero, Equation 4.3 switches signs.

To calculate  $N_Q$ , we create an initial guess of the minimum pre-watermarked coefficient. We set the minimum post-watermarked coefficient equal to  $t_o$  which is the minimum value we can quantize. We assume  $\text{modulo}(\text{coefficient}, S)$  will return  $S$ . This assumption gives a distortion larger than possible with the *modulo* function. We choose to embed a '1' because embedding a '1' adds  $\frac{S}{4}$  to the final output as

opposed to adding  $\frac{3}{4}S$  which embedding a '0' does. We add the smaller amount because we have just subtracted  $S$  from the coefficient, and we want to keep the post-watermarked value as distorted as possible from the original coefficient. By using the most distorted value, we calculate the largest minimum coefficient. This minimum coefficient is at least  $S - \frac{S}{4} = \frac{3}{4}S$  away from its original value. When we use actual minimum coefficients, they cannot be distorted this much.  $Modulo(\text{coefficient}, S)$  will always be less than  $S$ . Equation 4.4 gives the calculation of this initial minimum coefficient.

$$\text{initial minimum coefficient} = t_o + S - \frac{S}{4} = t_o + S\frac{3}{4} \quad (4.4)$$

We sort the image coefficients by their magnitude with greatest magnitude first. We search through the sorted image coefficients to find the minimum wavelet image coefficient just larger than this initial minimum coefficient. This gives us  $N'_Q$ , the starting  $N_Q$  value.

To find  $N_Q$ , we start with the coefficient  $N'_Q + 1$  specifies. Using this coefficient, we calculate its post-watermarked value as in Equation 4.5 again assuming embedding a '1.'

$$\text{post-watermarked coefficient} = \text{coefficient} - modulo(\text{coefficient}, S) + \frac{S}{4} \quad (4.5)$$

The difference with this calculation and the calculation for the initial minimum coefficient in Equation 4.4 is now we are no longer assuming  $modulo(\text{coefficient}, S) = S$ . If this post-watermarked coefficient is greater than  $t_o$ , then we increment  $N'_Q$  by one and take the next coefficient and repeat. When we find a coefficient whose post-watermarked value is less than  $t_o$ , we stop.  $N_Q = N'_Q$ . The algorithm in Table 4.1 demonstrates this iterative process.

For a constant  $T$  and  $Q$ , as  $S$  increases,  $N'_Q$  decreases. When calculating the true  $N_Q$ , as  $S$  increases,  $N_Q$  may or may not increase. This is because we are including the *modulo* function which is non-linear.  $Modulo(\text{coefficient}, S)$  can only return values between 0 and  $S$ . It does not return linear results for as the coefficient

```

 $N'_Q$  = number of coefficients larger than initial minimum coefficient
 $n_{temp} = N'_Q$ 
toosmall=false
while(toosmall=false)

    post-watermarked coefficient = coefficient of image at ( $n_{temp} + 1$ ) - modulo(coefficient of image at ( $n_{temp} + 1$ ),  $S$ ) +  $\frac{S}{4}$ 
    if (post-watermarked coefficient <  $t_o$ )
        toosmall = true
         $n_{temp} = n_{temp} - 1$ 
    end if
     $n_{temp} = n_{temp} + 1$ 
end while
 $N_Q = n_{temp}$ 

```

Table 4.1: Algorithm for determining  $N_Q$

approaches a multiple of  $S$ ,  $\text{modulo}(\text{coefficient}, S)$  approaches  $S$ . However, when the coefficient equals a multiple of  $S$ ,  $\text{modulo}(\text{coefficient}, S)$  equals 0. A coefficient may be too small for a subset of an  $S$  range with a given  $T$  and  $Q$ . It will work for the large and small  $S$  values but not for some in between values. An example follows.

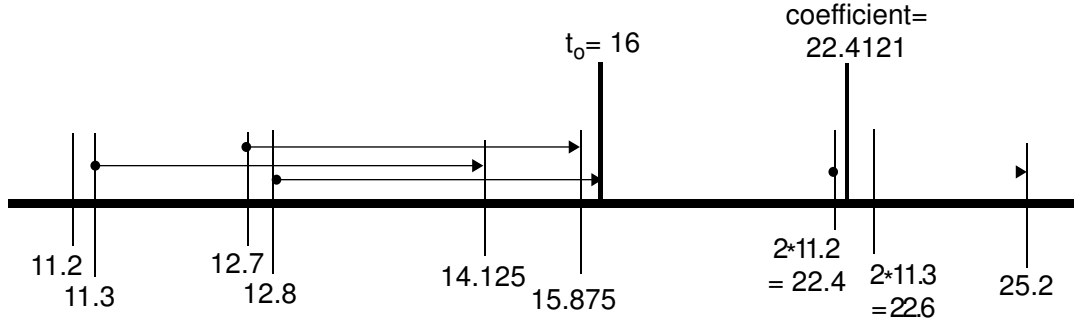


Figure 4.4: This figure shows how the modulo math changes the acceptable minimum coefficient over a range of  $S$ . For the coefficient=22.4121, the range  $11.2 \leq S \leq 12.7$  is unacceptable while  $S$  values outside this range are acceptable.

Figure 4.4 demonstrates how one coefficient is too small for a subset of  $S$ . In this example,  $T = 12$  and  $Q = 9$ , with  $t_o = 16$  from Equation 3.2. For an



$N_Q$  that gives a minimum coefficient=22.4121,  $S$  values between 11.3 and 12.7 are unacceptable. For  $S=12.7$ ,  $22.4121 - \text{modulo}(22.4121, S) = 12.7$ . Adding  $\frac{S}{4} = 3.175$  gives a post-watermarked value of 15.875 which is less than  $t_o$  and so will not be quantizable.

As  $S$  decreases, the post-watermarked value also decreases till  $S = 11.2$ .  $11.2 * 2$  gives 22.4 which is just less than the minimum coefficient, 22.4121. Therefore,  $22.4121 - \text{modulo}(22.4121, S) = 22.4$ . Before adding the  $\frac{S}{4}$ , we are already greater than  $t_o$ . Which makes the  $N_Q$  that allows this minimum coefficient acceptable for this  $S$ .

Moving in the other direction, allowing  $S$  to equal 12.8 makes the  $N_Q$  that gives us 22.4121 acceptable too.  $22.4121 - \text{modulo}(22.4121, S)$  is 12.8. Adding  $\frac{S}{4}$  which is 3.2 to 12.8 gives 16 which is our  $t_o$  making this quantizable and  $S$  acceptable.

Given a  $T$ ,  $Q$ , and  $S$ , we can specify the number of coefficients,  $N_Q$ , from the image we can use for watermarking. We sort the image coefficients in decreasing magnitude order. Using Equation 4.4, we calculate the initial minimum coefficient. Following the algorithm described previously, we test each coefficient in the image that is smaller than this initial minimum coefficient to find their post-watermarked value. When we find a post-watermarked value less than  $t_o$ , we have gone too far. Taking the index of the last coefficient whose post-watermarked value was not less than  $t_o$ , we have our  $N_Q$  for the given parameters.

#### 4.3 Given $T$ , $S$ , and $N$ , choose an Error Correction Code to meet the required IBER

By incorporating a BCH error correction code (ECC), we can lower the IBER to a rate acceptable to the user. As explained in Section 2.8, using the ECC entails a cost. Without ECC, every watermarked bit is an information bit. We can send up to  $N$  information bits with  $N$  being the number of coefficients we are using. However, when we incorporate ECC, a portion of the  $N$  watermark bits must be used for the coding. Using an  $n, k$  BCH ECC,  $\frac{k}{n}N$  bits are available as information bits. With

the ECC, we can no longer use all  $N$  watermark bits for information. We can only use a percentage of them because of the overhead in the ECC.

*4.3.1 Choosing  $N$ ,  $n$ ,  $k$ , and  $M$ .* When choosing the  $k$  and  $n$  values for the ECC, we must be aware of the impact on redefining  $M$  and  $N$ . The system determines  $N$  and  $M$  from the user's specified image and information bit stream. Independent of the choice for  $N$  and  $M$ ,  $n$  and  $k$  can also be set. Equation 4.6 shows the relationship between these variables.

$$M \leq \frac{k}{n}N \quad (4.6)$$

During the encoding step prior to watermarking, we break the  $M$  information bits into blocks of  $k$  length bits. These pass through the BCH encoder for mapping from  $k$  length blocks into  $n$  length codewords. If  $M$  is not a multiple of  $k$  we need to include extra bits into the information bit stream to ensure each block of bits is  $k$  length before going to the encoder.

In watermarking, we take the codewords of  $n$  bit length, append them together into one long watermark bit stream, and embed them into the  $N$  coefficients. The number of watermark bits is a multiple of  $n$ . We need to ensure that this is less than or equal to the number of coefficients,  $N$ . If the number of watermark bits is less than  $N$ , we need to append extra bits to the watermark bits to ensure that every coefficient receives a watermark bit. If we do not, then upon extraction, the system extracts a bit from every coefficient regardless of whether we embedded one. The system extracts bits from coefficients that never had bits embedded into them. These non-embedded bits would be extra bits that have no meaning in our extracted watermark. To ensure we do not end up with any extra bits, we encode a bit into every coefficient.

Since  $N$  and  $M$  are in general chosen independently, they may not be multiples of  $n$  and  $k$ , respectively. Thus, dummy bits must be embedded into the coefficients.

These dummy bits could be used to convey additional information or provide more error correction. These are lost opportunities for we are transmitting coefficients that only carry dummy bits. These superfluous coefficients can be removed which would also remove the need to include dummy bits. Therefore, in this research, we force  $N$  to be a multiple of  $n$ . We also chose  $M$  from the selection of  $N$ ,  $n$ , and  $k$  using Equation 4.7 which guarantees that  $M$  is a multiple of  $k$ , and we are using every bit possible for information storage for the given number of coefficients and ECC strength.

$$M = \frac{k}{n}N \quad (4.7)$$

*4.3.2 Binary Symmetric Channel.* The reason we are introducing ECC is because the user specified an IBER which we have not been able to meet with our previously selected parameters. We already have an experimental IBER that is too high. Equation 4.8 predicts the new IBER from incorporating the specified  $n, k$  ECC [13]. In this equation, the new IBER is  $P_{\text{ECC}}$  with  $p$  being the pre-ECC IBER.  $n$  is from our specified  $n, k$  code as is  $t$ , the number of bits in each codeword the ECC can correct. To use this equation as a predictor, our system must follow the properties of a binary symmetric channel.

$$P_{\text{ECC}} = \frac{1}{n} \sum_{j=t+1}^n j \binom{n}{j} p^j (1-p)^{n-j} \quad (4.8)$$

A binary symmetric channel states that the probability a transmitted symbol is received incorrectly is equal for the entire set of possible transmitted symbols. The probability that the symbol will be received incorrectly is the same regardless of the sent symbol. In our system, a binary symmetric channel implies that it is equally probable that an embedded ‘1’ will be incorrectly extracted as a ‘0’ as it is probable that an embedded ‘0’ will be incorrectly extracted as a ‘1.’ Figure 4.5 shows how for  $T = 11$ ,  $Q = 8$ , the system is binary symmetric for most values of  $S$  over the region

$\frac{t_o}{2}$  to  $t_o$ . The percentage of the total IBER of ‘1’s being extracted as ‘0’s is similar to the percentage of ‘0’s being extracted as ‘1’s.

Two values of  $S$  are not binary symmetric in our example. These occur at  $S = \frac{3}{4}t_o = 12$ , and  $S = \frac{t_o}{2} = 8$ . In these two instances, all the error is one way error. As  $S$  approaches  $t_o$  all the error also appears one way. However, because there are significantly fewer errors in this area, most of the errors appear within the same bad bin, and so have the same one way error. We consider these values of  $S$  to still cause a binary symmetric channel because the error is so limited.

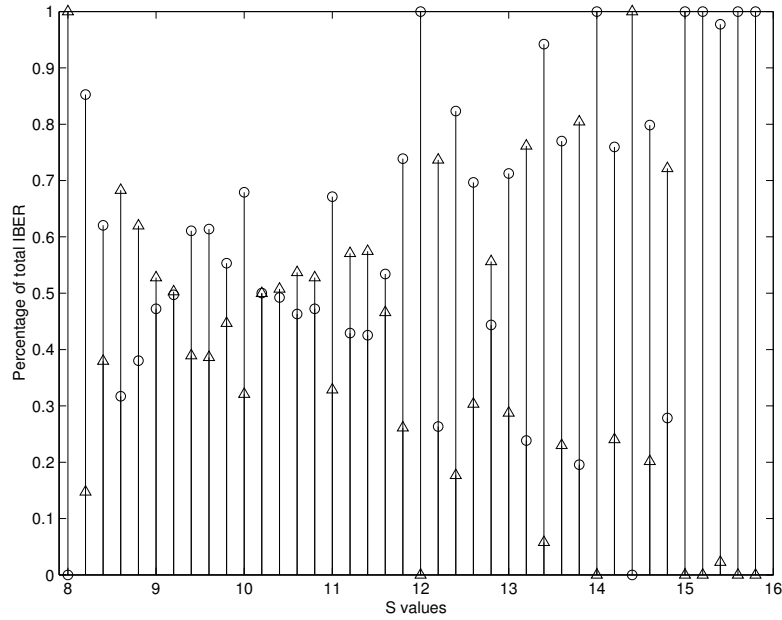


Figure 4.5: For  $T = 11$  and  $Q = 8$ , the system is binary symmetric for most values of  $S$ . The circles are the percentage of total IBER of ‘0’s being extracted as ‘1’s. The triangles are the percentage of ‘1’s being extracted as ‘0’s. For  $S = \frac{t_o}{2} = 8$  and  $S = \frac{3}{4}t_o = 12$ , all the error is caused by one way. The system is not binary symmetric for these cases.

For  $S = \frac{t_o}{2}$ , each bin contains an embedded ‘1’ and an embedded ‘0’ as demonstrated in Figure 4.2 earlier. Taking *modulo*(reconstructed value,  $S$ ) always returns  $\frac{S}{2}$ . Because  $\frac{S}{2}$  is greater than or equal to  $\frac{S}{2}$ , the output is always a ‘0.’ Therefore, all embedded ‘0’s are extracted as ‘0’s while all embedded ‘1’s are also extracted as ‘0’s. This causes a non-symmetric error channel for this  $S$  value.

When  $S = \frac{3}{4}t_o$ , we also have a non-symmetric error channel. For this  $S$  value, the bins are in a three bin repeated cycle. The first bin contains an embedded ‘1’ and extracts it as such when reconstructed. The second bin does the same for an embedded ‘0.’ The third bin contains both an embedded ‘1’ and an embedded ‘0.’ The reconstructed value in this third bin modulated with  $S$  is always 0. During the extraction step, because 0 is less than  $\frac{S}{2}$ , we always get a ‘1.’ This means that embedded ‘1’s will always be extracted correctly. Embedded ‘0’s will only be extracted correctly half the time because the third bin in the three bin repeated cycle always returns the incorrect bit while the second bin returns the correct bit. Therefore, all errors with this  $S$  are caused only from embedded ‘0’s being extracted as ‘1’s again causing a non-symmetric error channel.

Over most of the range of  $S$  values, the system is binary symmetric. Because of this, we can use Equation 4.8 to predict the IBER after incorporating an  $n, k$  ECC. For two values of  $S$ , however, we do not have a binary symmetric channel. We cannot predict the IBER when incorporating the ECC around these two  $S$  values .

*4.3.3 Error Correction Code Results.* Figure 4.6 shows the IBER for a constant  $Q$  and  $T$  with varying  $S$ . Each plot has a different  $n, k$  BCH code. The horizontal line is the error rate the given code can correct. This number is  $\frac{t}{n}$ . For  $S$  values with IBER below this correctable limit, the results with the ECC are lower than without it. For some instances, the IBER drops to zero.

Table 4.2 shows the overhead cost in terms of percentage of useable watermarking bits for each  $n, k$  pair in Figure 4.6. As the overhead increases, the percentage

$n$	$k$	overhead cost
63	10	$\frac{63-10}{63} = 84\%$
63	18	$\frac{63-18}{63} = 71\%$
31	16	$\frac{31-16}{31} = 48\%$
255	131	$\frac{255-131}{255} = 49\%$

Table 4.2: Overhead cost for using BCH error correcting code.

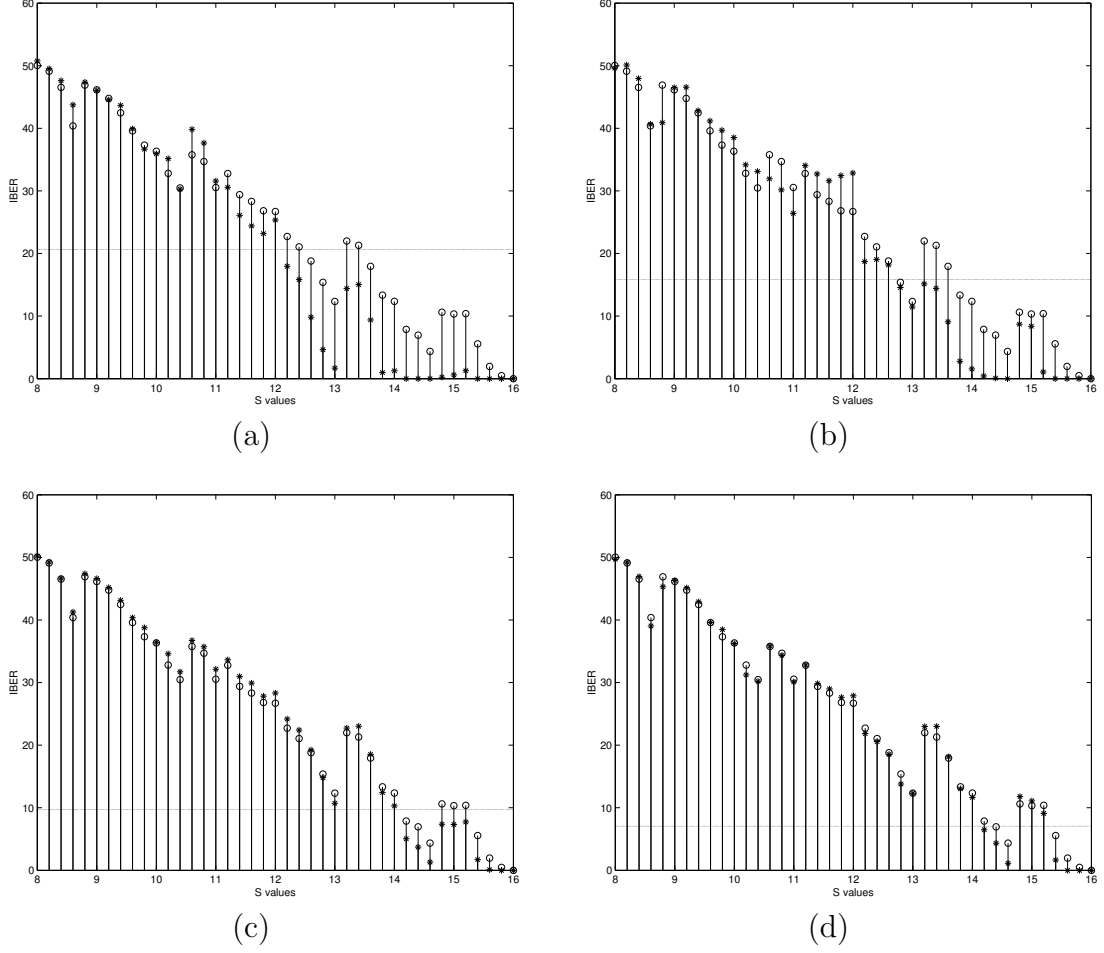


Figure 4.6: Plots showing the results of using BCH error correction codes with  $T = 11$  and  $Q = 8$  but different  $n, k$  combination. The circles are the IBER without any ECC, while the stars are those with ECC. When the IBER without ECC drops below the correctable line ( $\frac{t}{n}$ ), the corresponding IBER with ECC drops, sometimes to zero. (a)  $n = 63, k = 10$ . (b)  $n = 63, k = 18$ . (c)  $n = 31, k = 16$ . (d)  $n = 255, k = 131$ .

of bits corrected also increases. This allows us to use a weaker  $S$  while increasing the ECC. For example, if a user requires a high PSNR with a specific IBER, we would need to use a small  $S$  which increases the probability of extracted bit errors to keep a high PSNR. However, if we are able to use more image coefficients,  $N$ , for watermarking, we can incorporate a large ECC to correct these extracted bit errors while keeping  $S$  small and PSNR high.

Places that the experimental results and the predictive results do not match are around  $S = \frac{3}{4}t_o = 12$ . The system is not a binary symmetric channel for this value of  $S$  as explained earlier meaning Equation 4.8 does not hold. This shows that given an IBER and  $n, k$  values, we can predict the new IBER before implementing. We can predict the new IBER when using the BCH ECC.

#### 4.4 *Given $T$ , choose an $N$ , $Q$ , and $S$ to give the desired PSNR*

There is no prediction for PSNR. Equation 2.1 shows how the PSNR depends upon the mean square error between the original and reconstructed coefficients. The mean square error between these two values cannot be predicted because of the modulo arithmetic used in the reconstruction. This mean square error can only be known using the exact input coefficients. The experimental results dictate that PSNR relies heavily upon  $N$ : the larger the  $N$ , the higher the PSNR. We also know that  $N$  relies upon  $N_Q$  with  $N_Q$  relying upon  $T$ ,  $S$ , and more importantly,  $Q$  as seen in Section 4.2. Even though we cannot get a prediction for PSNR, we do see trends for PSNR based upon the parameters  $N$ ,  $Q$ , and  $S$ .

The biggest impact on PSNR is the number of wavelet coefficients and the quantization level. The quantization level,  $Q$ , determines to what precision the reconstructed values will be similar to the original. As  $Q$  increases, we store more data for each coefficient and so get a better reconstruction of the original value. However,  $Q$  specifies the maximum number of coefficients,  $N_Q$ , we can use from the image. Increasing the number of coefficients used in reconstruction has a more significant impact upon the PSNR than increasing the precision while using only a

few coefficients. Once  $Q$  is specified, the number of coefficients,  $N$ , becomes the most significant parameter. Because the image is transformed into the wavelet domain and the wavelet transform is parsimonious, we only require a small portion of the coefficients to reconstruct a high quality image. However, the more coefficients we have, the quality does improve. Figure 4.7 shows the PSNR with a constant  $T$ ,  $Q$ , and  $S$  while varying  $N$  to demonstrate the impact  $N$  has upon PSNR.

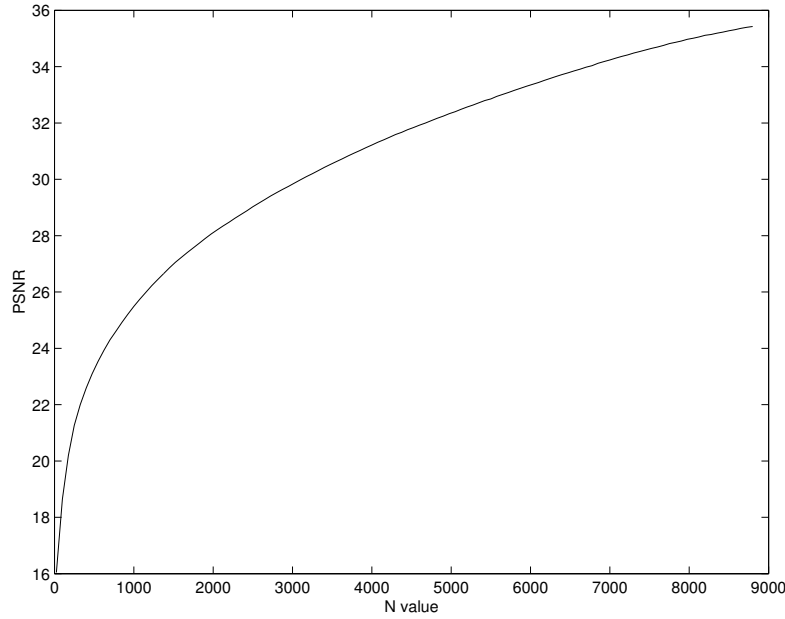


Figure 4.7: *The biggest impact on PSNR is the number of image coefficients available for reconstruction for a given a number of quantization iterations  $Q$ . Keeping  $S$  constant, the above plot shows the PSNR versus a varying  $N$ .*

Figure 4.8 shows the contour lines of constant PSNR for the specific  $S$  and  $Q$ . For this figure, the PSNR varies 9.2dB.  $N$  is chosen to equal the  $N_Q$  specified by the  $S$ ,  $Q$  pair. For a small  $Q$ , the  $N_Q$  is small, and  $S$  has little impact upon the PSNR as seen by the relatively smooth vertical contour line. As  $Q$  increases,  $S$  does seem to have more of an impact. For the range  $8 \leq S \leq 13$ , the  $Q$  value needed to keep a constant PSNR needs to increase. As we increase  $S$  in this region, we are required to save more bits to keep a constant PSNR.  $S$  only seems to impact the PSNR because the  $S$  value is directly affecting the  $N_Q$  which is the  $N$  for this PSNR



calculation. This change in  $N_Q$  because of the  $S$  is explained by the modulo function in calculating the post-watermarked values as in the example pictured by Figure 4.4 previously.

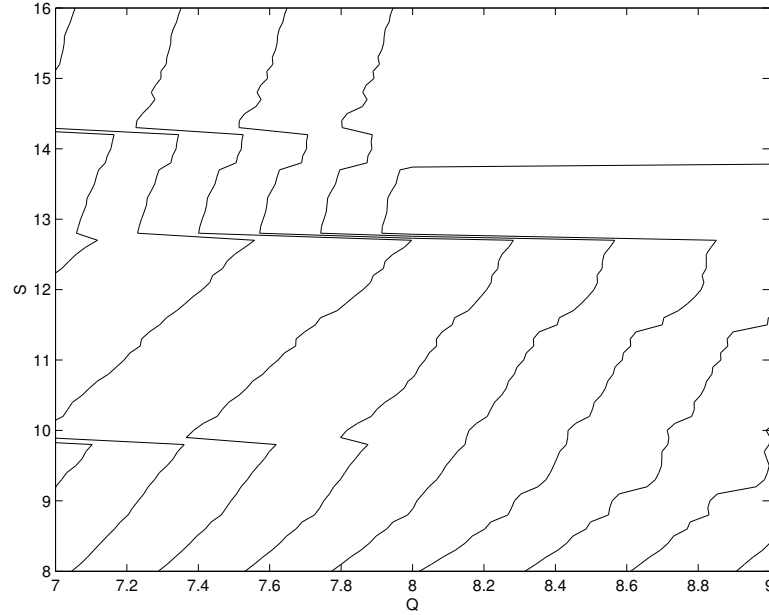


Figure 4.8: Contour lines of constant PSNR for the varying  $S$  and  $Q$ .  $T$  is constant.  $N$  equals the  $N_Q$  for the  $S, Q$  pair.

Forcing a small constant  $N$  for all the values of  $S$  and  $Q$  returns a range of PSNR values less than 0.05dB. If we increase  $N$  to a larger constant size as in Figure 4.9, we get a range of 0.3dB, six times the size previously. However, we can no longer use some of the  $S, Q$  pairs because this larger  $N$  exceeds the  $N_Q$  for these pairs. From Equation 3.3, to stay out of the *Dropped Bits* region,  $N$  must be less than or equal to  $N_Q$ .

We do not have a prediction for PSNR. We do see trends though. Using a  $Q$  value that offers a large enough  $N_Q$ , the larger the number of coefficients,  $N$ , we use, the higher the PSNR. Also, the strength of watermarking,  $S$ , impacts the choice of  $N$ . For some ranges of  $S$ , the  $N_Q$  is limited to a smaller value because of the modulo arithmetic in watermarking the coefficients.

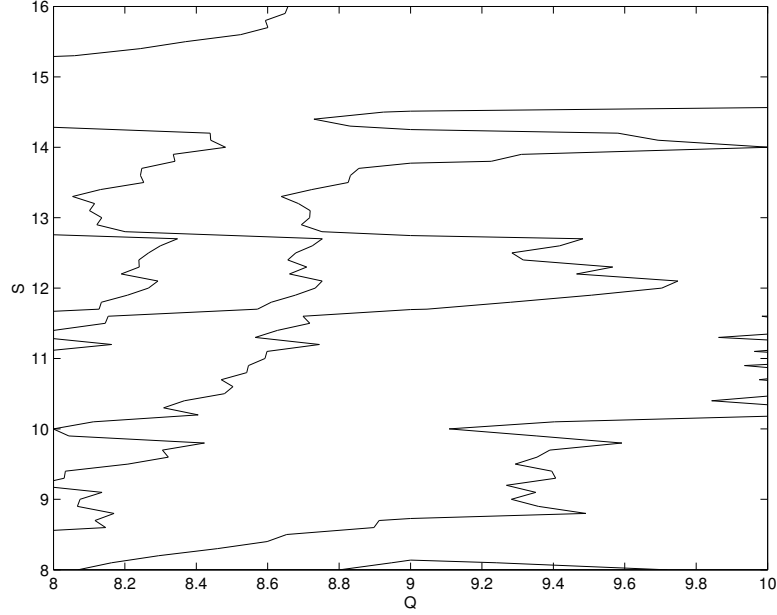


Figure 4.9: For a large constant  $N$ , PSNR varies over the range of  $S$  and  $Q$ . However, some  $S, Q$  pairs cannot be used as their  $N_Q$  is smaller than the large  $N$ .

#### 4.5 Example Output

Figure 4.10 shows an example output of this research. Figure 4.10(a) is the original image without any compression. Figure 4.10(b) shows the reconstructed image embedding  $M = 8,160$  information bits using the same number of coefficients,  $Q = 9$  quantization iterations, and an embedding strength of  $S = 7$ . We get a PSNR of 35.5dB, an IBER of 6.8%, and a bit rate of  $\frac{51,355}{256^2} = 0.78$ . Our theoretical IBER calculation with these parameters is 12.75%. By incorporating a 255,131 BCH ECC, Figure 4.10(c) shows the image maintains a quality PSNR, 35.5dB. We maintain a bit rate of 0.78. By using the ECC, we lower our IBER from 6.8% to 2.2%. The cost is a reduction in the number of coefficients actually carrying information bits. Incorporating the ECC, we only embed  $M = 4,192$  information bits. Only  $\frac{4,192}{8,160} = 51\%$  of the coefficients carry information. Without ECC, we embed only the information bits, but with the ECC we embed the coded bits. Our system is viable for we are able to achieve the user specified IBER by varying different parameters without adversely affecting the PSNR.



(a)



(b)



(c)

Figure 4.10: (a) Shows the original image before any compression. The next two images show the reconstructed image using quantization iterations  $Q = 9$ , embedding strength  $S = 7$ , and number of coefficients  $N = 8,160$ . (b) Without using any error correction, we get an  $IBER=6.8\%$ , a  $PSNR=35.5dB$ , and embed  $M = 8,160$  information bits. (c) Incorporating a 255,131 BCH error correction code, we lower the  $IBER$  to  $2.2\%$  while maintaining a  $PSNR$  of  $35.5dB$ . The cost is lowering the amount of information bits we embed to  $M = 4,192$ .

## 4.6 *Summary*

This chapter explained how given goals by the user: reliability of transmitting the information bits (IBER), the number of information bits to transfer ( $M$ ), the bit rate, or the image quality (PSNR), we can determine the set of variables to operate the system. We performed a thorough analysis and provided examples to demonstrate this. Given a specific IBER and image, we know the relationship between the embedding strength and number of quantization iterations to select values to guarantee the IBER. We have also shown how to incorporate an error correction code to lower the IBER without increasing the embedding strength which has a negative impact on PSNR. For a specified bit rate or number of information bits to transfer, we demonstrated how we can choose the number of wavelet coefficients to use that achieves these goals without introducing information bit errors due to dropped bits. Finally, we explained that even though we cannot meet a specific PSNR, we can estimate the PSNR through trends in the number of wavelet coefficients used.

## V. Conclusions

### 5.1 Contributions

This work began by considering embedding audio information into video frames. However, through our frame-by-frame analysis and the nature of the digital watermarking technique fundamental to this system, any information stream can be embedded as the watermark. Possible other watermark streams include a copyright, a text message, or an image. Similarly, any digital medium can be the source we embed the watermark into. We are not restricted to audio nor are we restricted to video frames.

Through our analysis and experimental results, we have demonstrated via examples that the compression system for embedding binary data into video frames can be controlled by a set of variables. We demonstrated that given an information bit error rate (IBER) specified by the user, we can set the parameters of the system to achieve this requirement. By working in the *Extraction Error* region of the IBER plot, we can select the variables to achieve the specified IBER. We use this region of the IBER plot because operating in the *Dropped Bits* region gives 50% IBER which is not correctable. We demonstrated that in the *Perfection Extraction* region, IBER=0% thus, the variables are free to minimize other criteria: amount of information bits, bit rate, or PSNR.

Instead of requiring a specific IBER, the user may specify a number of information bits to embed or a bit rate to achieve. We explained how to take an image file and by varying the number of quantization iterations and embedding strength, we can find the maximum number of coefficients that can be quantized and reconstructed. By selecting a number of coefficients less than or equal to this maximum value and equal to the number of information bits, we can ensure that every information bit is embedded into a coefficient, the coefficient quantized and reconstructed which allows

the information bit to be extracted. By choosing a low number of coefficients for the given quantization iterations, we can achieve the bit rate requirement.

If the user wishes to decrease the IBER and is willing to pay the cost, an error correction code (ECC) can be incorporated into the system. Because we have demonstrated that the system is a binary symmetric channel over most embedding strengths, we can use an error correction code (ECC) and predict the new IBER. The cost for using the ECC comes in transmitting more watermark bits than information bits. With the ECC, we encode the  $k$  length information block into an  $n$  length codeword which we embed. Because  $n > k$ , we have more watermark bits than information bits as opposed to not using the ECC where the number of watermark bits equals the number of information bits.

We also analyzed the last requirement a user can specify, the PSNR. Even though we cannot predict the PSNR, we explained trends in the variables that impact the PSNR. The number of quantization iterations dictate how much precision we retain when we reconstruct the coefficients. The more precision, the lower the error leading to a higher PSNR. However, the real power of the number of quantization iterations in relation to PSNR comes in dictating the largest number of coefficients that can be used. By increasing the number of coefficients, the PSNR increases. Varying the number of coefficients retained impacts the PSNR more so than keeping the the number of coefficients constant and varying the embedding strength and number of quantization iterations.

In conclusion, our research shows that we can meet the user requirements for the state of the system. The state of the system can be viewed as the information bit error rate, the number of information bits to transmit, the bit rate, and the PSNR. Our experimental results demonstrate that our calculations are correct and provide a means for choosing optimal operating points in the “Wavelet-Based Audio Embedding & Audio/Video Compression” system of Mendenhall [10]. We analyzed a specific embedding and compression scheme. The analysis may apply to other

compression schemes (JPEG), however our results are tied to this specific technique. With the analysis and incorporation of free variables, we have increased the versatility of our technique making it more competitive as a viable compression scheme for video data.

## *5.2 Future Work*

The next step in the embedding/compression system evolution is automation. The automation can select the variable values to dictate the state of the system to meet the user requirements. If given less stringent requirements, the automation can return a range of values that meets the requirements for the user to choose from. This would allow the user to make a general requirement and upon iterations create more specific requirements with the aid of the automation.

A frame-by-frame analysis can be performed to achieve a constant transmission bit rate. By maintaining a constant bit rate, a bandwidth requirement can be met.

Using the three-dimensional nature of video, different frames may take more of the embedding load than others. Some frames may be able to mask more watermark bits easier than others. By adjusting the amount of watermark bits for each frame, we may be able to achieve a better overall compression ratio.

## Bibliography

1. Burrus, C. Sidney, et al. *Introduction To Wavelets and Wavelet Transforms: A Primer*. Englewood Cliffs, NJ: Prentice-Hall, 1997.
2. Claypoole, Roger L. *Adaptive wavelets transforms via lifting*. PhD dissertation, Rice University, Houston, Texas, October 1999.
3. Cox, Ingemar J., et al. "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Transactions on Image Processing*, 6(12):1673–1687 (1997).
4. Cox, Ingemar J., et al. *Digital Watermarking*. Morgan Kaufmann Publishers, 2002.
5. Hartun, F., et al. "Spread Spectrum Watermarking: Malicious Attacks and Counterattacks," *SPIE Conference on Security and Watermarking of Multimedia Content*, 2657:147–158 (1999).
6. Huang, Jiwa and Yun Q. Shi. "Adaptive Image Watermarking Scheme Based on Visual Masking," *Electronics Letters*, 34(8):748–750 (1998).
7. Kelly, D. H. "Visual Response to Time-dependent Stimuli: Amplitude Sensitivity Measurements," *Journal of the Optical Society of America A*, 51:422–429 (1961).
8. Langelaar, Gerhard C., et al. "Watermarking Digital Image and Video Data: A State-of-the-Art Overview," *IEEE Signal Processing Magazine*, 20–46 (2000).
9. Mallat, S. G. "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693 (1989).
10. Mendenhall, Michael J. *Wavelet-Based Audio Embedding & Audio/Video Compression*. MS thesis, Air Force Institute of Technology, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 2001.
11. Sayrol, E., et al. "Optimum Watermark Detection for Color Images," *IEEE International Conference on Image Processing*, 2:231–235 (1999).
12. Shapiro, Jerome M. "Embedded Image Encoding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, 41(12):3445–3462 (1993).
13. Sklar, Bernard. *Digital Communications: Fundamentals and Applications*. Englewood Cliffs, New Jersey 07632: PTR Prentice Hall, 1988.
14. Swanson, Mitchell D., et al. "Robust Audio Watermarking using Perceptual Masking," *Signal Processing*, 66:337–355 (1998).
15. Tian, Jun and JR. Raymond O. Wells. "A Lossy Image Codec Based on Index Coding." *IEEE DCC*. 1996.



16. Tsai, Min-Jen, et al. "Joint Wavelet and Spatial Transformation for Digital Watermarking," *IEEE Transactions on Consumer Electronics*, 46(1):241–245 (2000).
17. Turnbull, Bruce H. "Important Legal Developments Regarding Protection of Copyrighted Content Against Unauthorized Copying," *IEEE Communications Magazine*, 92–100 (2001).
18. Voyatzis, G. and I. Pitas. "Applications of Toral Automorphisms on Image Watermarking," *IEEE International Conference on Image Processing*, 2:237–240 (September 1996).
19. Wei, Z. H., et al. "Perceptual Digital Watermark of Images using Wavelet Transform," *IEEE Transactions on Consumer Electronics*, 44(4):1267–1272 (1998).
20. Wu, Chuan-Fu and Wen-Shyong Hsieh. "Digital Watermarking Using Zerotree of DCT," *IEEE Transactions on Consumer Electronics*, 46(1):87–94 (2000).
21. Yarbus, Alfred L. *Eye Movements and Vision*. 227 West 17 Street, New York, NY 10011: Plenum Press, 1967.
22. Zhang, Bo and Yuan F. Zheng. "Packed Integer Wavelet Transform Constructed by Lifting," *IEEE Transactions on Circuits and Systems for Video Technology* (2000).

## *Vita*

Brendan is a St. Louis native. He has degrees from schools and awards from work. He belongs to a couple of different organizations for academic and professional reasons. Brendan has set fire to his kitchen while cooking enchiladas in Texas and exploded a pot boiling water in Ohio. Undeterred, he still experiments in the kitchen with cooking and will most likely do something else unexpected in New Mexico.

He watches too much television and is obsessed with movies. When he's not annoying his neighbors with his poor guitar skills, he's listening to music by those who have talent. Reading too many books and desensitizing himself through video games are also ways Brendan wastes his time.

<b>REPORT DOCUMENTATION PAGE</b>				Form Approved OMB No. 074-0188		
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.						
<b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>						
<b>1. REPORT DATE (DD-MM-YYYY)</b> 25-03-2003		<b>2. REPORT TYPE</b> Master's Thesis		<b>3. DATES COVERED (From - To)</b> Aug 2001 - Mar 2003		
<b>4. TITLE AND SUBTITLE</b>  THEORETICAL ANALYSIS OF INFORMATION WATERMARKING IN WAVELET-BASED VIDEO COMPRESSION				<b>5a. CONTRACT NUMBER</b>		
				<b>5b. GRANT NUMBER</b>		
				<b>5c. PROGRAM ELEMENT NUMBER</b>		
<b>6. AUTHOR(S)</b>  Casey, Brendan K, Captain, USAF				<b>5d. PROJECT NUMBER</b>		
				<b>5e. TASK NUMBER</b>		
				<b>5f. WORK UNIT NUMBER</b>		
<b>7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)</b> Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-7765				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  AFIT/GE/ENG/03-07		
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> AFRL/IFTA Attn: Dr. Robert L. Ewing 2241 Avionics CI WPAFB OH 45433-7334				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>  Air Force Material Command (AFMC)  DSN: 785-6653, ext 3592 e-mail: Robert.Ewing@wpafb.af.mil		
<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>						
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b>  APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.						
<b>13. SUPPLEMENTARY NOTES</b>						
<b>14. ABSTRACT</b> Embedding audio bits into images for transmission of video data alleviates the synchronization problem common in video transmission techniques. We continue work combining audio or other information bits and images into one file using digital watermarking techniques to correct the synchronization problem. The system compresses the file by using wavelet image coefficients and implementing bit plane coding.  Our research encompasses incorporating five free variables into the watermark/compression technique. These variables are watermark robustness, number of coding iterations, number of image coefficients, number of watermarked information bits, and number of watermarked error correcting bits. By altering these variables, four measurements of the output change. The measurements are the information bit error rate, the image quality, the bit rate, and the amount of watermarked data. We theoretically demonstrate how the variables impact these measurements. Experimental results on real video data support our theoretical findings. By analyzing each video frame, an automated system is able to choose optimal values of the five variables to meet specified measurement constraints.						
<b>15. SUBJECT TERMS</b> Compression, Mathematical Analysis, Statistical Analysis, Information Systems, Information Processing, Data Compression, Information Transfer, Image Compression, Video Frames, Numerical Analysis, Wavelet Transforms, Data Transmission Systems						
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>	<b>18. NUMBER OF PAGES</b>	<b>19a. NAME OF RESPONSIBLE PERSON</b> Roger L. Claypoole, Jr., Maj, USAF (ENG)	
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	UU	67	<b>19b. TELEPHONE NUMBER (include area code)</b> (937) 255-6565, ext 4620; e-mail: Roger.Claypoole@afit.edu	